國 立 交 通 大 學

電信工程研究所

碩士論文

使用對齊單音 MIDI 改善流行歌曲旋律擷取

Aligning Popular Music with Mono MIDI for Singing Pitch Extraction

研 究 生 ：蔡昌祐

指導教授：陳信宏 博士

中華民國一百零二年四月

使用對齊單音 MIDI 改善流行歌曲旋律擷取

Aligning Popular Music with Mono MIDI for Singing Pitch Extraction

研究生：蔡昌祐　　Student: Chang-You Cai

指導教授：陳信宏 博士　　Advisor: Dr. Sin-Horng Chen

國立交通大學

電信工程研究所

碩士論文

A Thesis

Submitted to Institute of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Communication Engineering

April 2013

Hsinchu, Taiwan, Republic of China

中華民國一百零二年四月

使用對齊單音 MIDI 改善流行歌曲旋律擷取

研究生：蔡昌祐　　指導教授：陳信宏 博士

國立交通大學電信工程研究所碩士班

中文摘要

音高代表聲音的基本頻率，在 Query-by-Singing/Humming (QBSH)系統上是一個重要的特徵，利用此特徵值的比對找出最相似的歌曲為 QBSH 系統的主要方法，因此音高偵測的準確度變得相當重要。而流行歌曲中的歌唱音高可以用人耳辨識出來，但因為背景伴奏、諧波等干擾的緣故，要利用電腦算出流行歌曲的歌唱音高將會困難許多。

本論文首先參考前人所提出的音高擷取方法來計算出流行歌曲的音高曲線，此方法會先壓抑樂器伴奏來提高人聲能量，並用疊加諧波後的頻譜做一連串的處理，再使用一種人聲音高頻率範圍偵測的方法來消除諧波，最後使用動態規劃法來擷取出音高曲線。但此方法仍存在一些缺點，且音高曲線並沒有將非人聲段部分去除，因此本論文提出一種改善前人作法的方法，基本構想是使用單音 MIDI 與要處理的流行歌曲對齊來協助改善音高之偵測，首先計算出各自頻率刻度轉換後的頻譜，並建立相似矩陣做動態時軸校對以找出單音 MIDI 中每個音符對應到流行歌曲的時間，並使用一套後處理的方法來修正不自然的音符，最後利用對齊好的單音 MIDI 判斷人聲段與非人聲段，並重新計算更準確的音高曲線。實驗結果顯示，本方法可以有效改進流行歌曲的音高偵測。

Aligning Popular Music with Mono MIDI for Singing Pitch Extraction

Student: Chang-You Cai Advisor: Dr. Sin-Horng Chen

Institute of Communication Engineering
National Chiao Tung University

Abstract

Pitch represents the fundamental frequency of voice. It is an important feature in a Query by Singing or Humming (QBSH) system. Currently, using pitch feature to find the most matched song is a popular way in QBSH. The accuracy of pitch detection is hence a critical issue. Although human can recognize singing pitch in a song with music accompaniment, it is not easy for a computer to automatically detect the singing pitch from a song because of the inferences of background music and harmonics.

In this thesis, we first use an existing method to extract the melody line of a popular song. The method first depresses the background music to enhance the singing voice. It then uses a method to enhance the pitch signal by summing harmonics. A method to estimate the range of human's pitch is then applied to eliminate all harmonics. Lastly, it finds the melody line by dynamic programming. Some drawbacks of the method can still be found, including the inaccuracy of pitch tracking at the beginning of singing signal and the existence of melody line at the non-singing part. We hence propose a method to improve it in this study. The method uses the monophonic MIDI signal aligned with the processing song to help to improve the pitch detection. It first computes the MIDI scale spectra of the two signals and sets up a similarity matrix for their alignment. A post-processing is then employed to segment the song and detect unnatural notes. Lastly, it utilizes the aligned MIDI to determine the vocal (singing) segment of the song and recalculates the melody line. Experimental results confirmed the effectiveness of the proposed approach.

誌謝

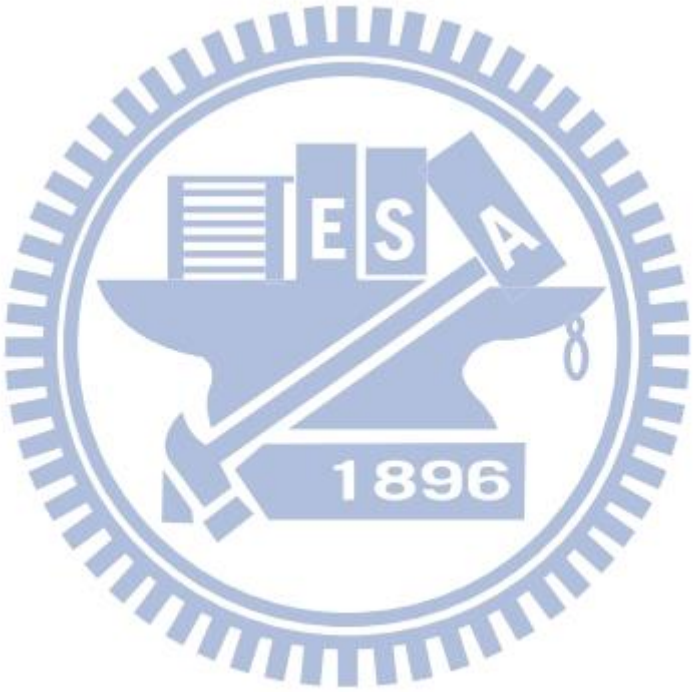
首先我要感謝陳信宏老師在我升碩二暑假時收留了我讓我進入707語音處理實驗室這個大家庭，而且老師真的是一個很認真的老師，平常除了要處理學校行政、上課之外還常常到實驗室關心我們的研究進度，深怕我們進度落後。而另外我還要感謝在中華電信研究所工作的廖宜彬學長，從一開始的懵懵懂懂讓我逐漸步上軌道，雖然我時常做的不好學長也不會太指責我，另外我們生日還是同一天。

此外，也感謝佳緯學長和我一起孤軍奮戰研究音訊處理這門學問。感謝小蝦學長每個禮拜都帶我們吃好吃的並常常上健身房鍛鍊肌肉為了將來的烏克蘭新娘做準備，不知道未來還能不能在台灣見到你。感謝和我一起轉進來籃球痛電余祥銓的小高以及不太熟的雅婷，希望你們趕快回來。感謝時常和我PK籃球的嬌娃與純情小助理靖觀，希望你們趕快結婚!!!感謝比我還宅的企鵝給我一些程式的建議並告訴我遊俠網的存在。感謝人生勝利組小邱、又帥又白的睿詮與可愛的俊翰，我不會忘記我們一起去台北的日子的!我還要感謝下一屆學弟妹，跑步超強的阿龐，酷似趙又廷的良基，最愛QQ的婉君，養蝦達人子睿，很有學問的奕勳，我早你們一步先離開了。而我的下下屆學弟妹，大老闆愛將聲鋒，籃球戰車ML茂隆，熱愛棒壘球的王柏，籃球超強的阿駿，唱將阿璋，資深宅男蔡仲堯以及非常神秘的李佩樺，期許你們明年能順利畢業。

最後我要感謝我的父母與家人，謝謝你們一路上的支持讓我能夠堅持下去完成碩士學位。

目錄

表目錄



圖目錄

第一章 緒論

1.1 研究動機

由於自然語言處理(natural language processing)技術的迅速發展，文字內容檢索系統(text content-based information retrieval system)已十分普遍，我們知道現今 Google 是一個很強大的搜尋引擎，只要我們有什麼疑問都能夠透過輸入關鍵字的方式，在網路上找出問題的答案或是要尋找的東西。而現今音樂也是一個日益龐大且迅速成長的資料族群，但是相較於文字資料庫，還尚未有相同成熟的音樂資料庫開始發展。

過去音樂資訊的查詢，通常是以文字的方式來達成，例如將一個音樂的資料以作曲者、歌名或演唱者等方式儲存在資料庫中，但是搜尋者通常不會記得歌名或是演唱者的姓名，反而比較會記得片段的旋律或是片段的節奏，因此開發一個藉由輸入片段音樂來搜尋的系統是有需要的，如用敲打來查詢（query by tapping）、以哼唱來查詢（query by humming）等方式。

本篇論文的主題在探討哼唱查詢(Query by Singing/Humming, QBSH)的特徵值擷取，通常 QBSH 系統會比較輸入端與資料庫中的特徵值相似度，進而找出最匹配的歌曲，因此特徵值的準確度相當重要。在音樂領域中常見的特徵值有色度(chroma)、音高(pitch)等等，音高與色度的差別主要是在色度忽略了諧波(harmonic)的影響，因此不會有高八度或是低八度的問題，但這也降低了特徵值的強健性。本篇論文以流行歌曲的音高擷取為重點，考慮樂器伴奏、諧波、及人聲段判斷對其影響的問題。

1.2 文獻回顧

早在 1995 年，Asif Ghias、Jonathan Logan、David Chamberlin 以及 Brian C. Smith 就在 ACM 多媒體研討會中發表了一套名為 QBH (Query By Humming)的系統[1]，此系統可以讓使用者藉由麥克風哼唱一小段音樂，搜尋資料庫中最相似的歌曲。他們主要是

透過自相關演算法(Auto-correlation method)求出使用者的音高曲線，並將音高曲線轉成U（這個音比前一個音高)、D(這個音比前一個音低)、R(這個音和前一個音相同)3 種字元組成的序列來找出資料庫中最相似的歌曲。然而此系統並未達到全面自動化的功能，使用者還需自行切割音符，且特徵值的準確率還有待提升。

自從 1999 年 Goto[2]首次使用統計的方法訓練參數模型來擷取音高曲線之後，有愈來愈多機器學習的方法被投入在旋律擷取的研究中，然而若是訓練的資料太少，旋律擷取的準確度會大打折扣。除了機器學習的方法外，也有一些研究投注在頻譜分析的方法上，例如 Hsu 等人[3]在 2010 年的 Music Information Retrieval Evaluation eXchange (MIREX) 當中使用了一連串的頻譜分析演算法，首先它壓抑樂器伴奏並加強頻譜上歌唱音高的部分，接著使用了一套人聲音高頻率範圍偵測的技巧來消除諧波的部分，最後以動態規劃的方法取得歌唱音高。

而在音符對齊方面，對齊的演算法主要有兩種：Dynamic Time Wrapping (DTW)與Hidden Markov Model (HMM) [4]，一般而言 DTW 比 HMM 簡單而容易實踐，且不用事先訓練模型。在[5]的研究中，採用先對 MIDI 重新合成的領航音檔與流行歌曲各自計算頻譜，在建立相似矩陣之後使用 DTW 演算法來做特徵值的時間對齊，而對齊好的 MIDI 就當作流行歌曲的人聲音高答案，實驗顯示對齊的準確度約有 80%。

1.3 研究方向

如何在輸入端輸入片段歌曲來找出資料庫中最匹配的歌曲為 QBSH 系統的最終目的。圖 1.1 為一般歌曲搜尋系統流程圖，通常資料庫的音樂形式分為兩種，一種為樂譜的格式，如 MIDI 和 Humdrum 等，另一種則是真實的錄音檔案，如 CD 音樂和 mp3 等，通常前者特徵值會較容易擷取，擷取出特徵值後就可以透過一些比對演算法算出相似歌曲的排名。

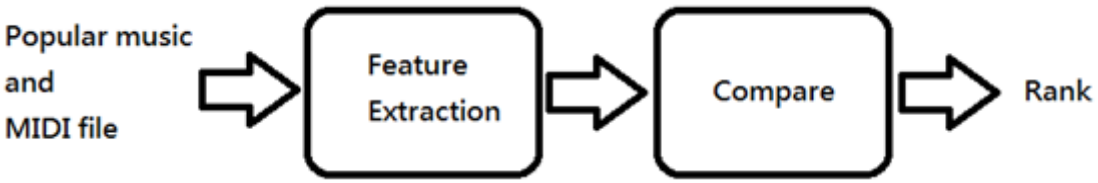


圖 1.1 歌曲搜尋流程圖

通常會假設使用者在哼唱時環境是安靜無聲的，因此要擷取出使用者的特徵值並不會太難，然而資料庫中的流行歌曲錄音檔案通常有背景伴奏，如何處理這類錄音檔案，忽略其背景伴奏而正確地擷取出演唱者的特徵值為本論文研究重點，只要流行歌曲的人聲歌唱音高擷取的愈準確就愈能夠找到我們想要的歌曲。

1.4 章節概要

本論文一共分為六章，各章節的內容分配如下:

第一章　緒論：介紹本論文之研究動機與方向。

第二章　單音 MIDI 資料庫與流行歌曲介紹：說明 MIDI 與一般流行歌曲的特徵，以及 MIDI 資料庫的特性。

第三章　音高擷取方法：Hsu 的音高擷取方法介紹。

第四章　使用單音 MIDI 改善音高曲線擷取：單音 MIDI 與流行歌曲對齊方法介紹，並重新計算音高曲線。

第五章　實驗結果與分析：結合對齊後的 MIDI 所求出的音高曲線之實驗結果。

第六章　結論與未來展望。

第二章 單音 MIDI 資料庫與流行歌曲介紹

本章說明 MIDI 資料庫的特性，以及和一般流行歌曲的差別，第四章將利用此 MIDI 資料庫的特性做對齊的後處理，對齊好後將利用兩者的特性來提高旋律擷取的準確度。

2.1 MIDI 與流行歌曲的優缺點

目前音樂內容檢索技術依照處理的對象主要分為兩種，一種是樂譜的形式，如 MIDI、Humdrum 等，另一種為真實的錄音檔案，如 CD 音樂、mp3 等。兩種類型的音樂各有其優勢與不足的地方，舉例來說 MIDI 所記錄的並不是聲音訊號，而是音符、音長、音量等控制參數，在擷取音高、音量、節奏等會較容易，因此在旋律擷取、和弦檢測等研究領域中，前者往往效果較後者好，且 MIDI 有占用記憶體少、製作成本低、方便編輯等優點，但與 CD 音樂、mp3 等音訊格式相比，MIDI 的音高欠缺真實感，每個音高呈現的方式非常平整，且音高變化不平滑，與自然的演唱有一定的差距，因此無法使用於音樂情緒分類、歌者辨識等研究。而在對流行音樂做旋律擷取之研究中，因為有樂器伴奏的干擾，以及諧波的問題，目前依然有相當大的難度，本論文將藉由對齊這兩個類型的音樂訊號，利用它們各自的特性，做增進音高曲線擷取準確度的探討。

2.2 MIDI 資料庫來源以及流行歌曲

MIDI 音樂的製作是看著樂譜把旋律用鍵盤(Keyboard)彈到電腦裡，因此有音高和音長等資訊，並利用音源器，選擇該有的音色，MIDI 製作的過程，可以參考[6]。目前的 MIDI 音樂資料庫(MIDI_888)是從網路上取得的，裡面總共有 888 首歌曲，這 888 首歌曲只記錄了人聲主旋律的部份，忽略樂器伴奏的部分，為單音的歌；有些原曲若為合

唱，則會有兩個版本，如版本 1 記錄男聲，版本 2 則記錄女聲；或是有人聲發出的伴奏(Rap)，也會分成兩個版本。因此，此資料庫總共有 1033 首歌。

下表是隨機選出的 10 首流行歌曲，比對 MIDI_888 中的單音 MIDI 音樂檔來觀察資料庫的特性。

表 2.1 MIDI_888 資料庫特性

歌曲	特性
廣島之戀	MIDI 檔整體約快 3 秒。此首歌為合唱，版本 1 記錄男聲，版本 2 記錄女聲
江南	MIDI 檔整體約慢 16 秒左右，與原流行歌曲間奏誤差大，有人唱的伴奏
小薇	MIDI 檔整體快了 3 秒左右
我們的愛	MIDI 檔整體快 9 秒左右，有人唱的伴奏
十年	MIDI 檔整體快 1 秒左右，間奏有落差
勇氣	MIDI 檔整體快 1 秒左右
聽海	MIDI 檔整體慢 10 秒左右
髮如雪	MIDI 檔整體快 24 秒左右，版本 2 有人唱的伴奏
屋頂	MIDI 檔整體快 6 秒左右，MIDI 版本 1 記錄男聲版本 2 記錄女聲，間奏有落差
心如刀割	MIDI 檔整體快 2 秒左右

由上表可以發現，此資料庫大致上是副歌、主歌開始的時間不準，間奏的長短也不一定符合要對齊的歌曲，但是在副歌或是主歌裡的音符相對位置大致上是準確的，之後將利用此特性來做後處理。目前僅考慮只有一個版本的 MIDI，而要對齊的流行歌曲是用[7]所提供的 You- Tube 轉 mp3 的技術，從 YouTube 找出非 MV 版本的歌曲，並一律轉成單聲道，取樣頻率為 16*k*Hz，解析度為 16-bit 的 wav 檔。

2.3 MIDI 歌曲檔說明

目前由 MIDI 歌曲能擷取出的資訊有 pitch、velocity、onset time in seconds、duration in seconds 等，以下擷取一段 MIDI 來做說明

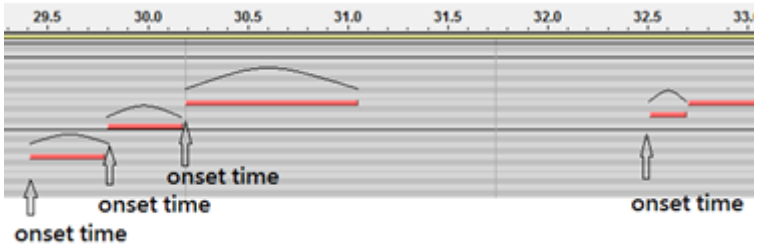


圖 2.1 單音 MIDI 歌曲示意圖

圖 2.1 中數字的單位為秒，這裡總共有 5 個音符，每個音符的高度就是音高，箭頭所指的地方表示音符起頭音的時間位置，符號∩表示音符的長度。有些 MIDI 檔中音符可能會發生一小部分的重疊，若發生重疊，將會把發生重疊的音符長度縮短到下一個音符的起頭音(onset time)位置，讓它不要有重疊發生。

第三章 Hsu 的音高擷取方法

Hsu 的音高擷取方法是 2010 年的 MIREX 中辨識率最高的方法，它主要是針對一般錄音音檔所設計，首先它壓抑樂器伴奏以提高人聲能量，接著計算出頻譜，為了增加有音高與沒有音高的能量差距，它使用了疊加諧波的技巧重新計算頻譜，而諧波的問題會使用一套人聲音高頻率範圍偵測的方法，找出頻譜上人聲音高頻率的範圍，最後使用基於動態規劃(dynamic programming)的方法擷取出音高曲線。

3.1 系統流程圖

此方法主要是針對頻譜的特性做一連串的處理，圖 3.1 為此方法的流程圖

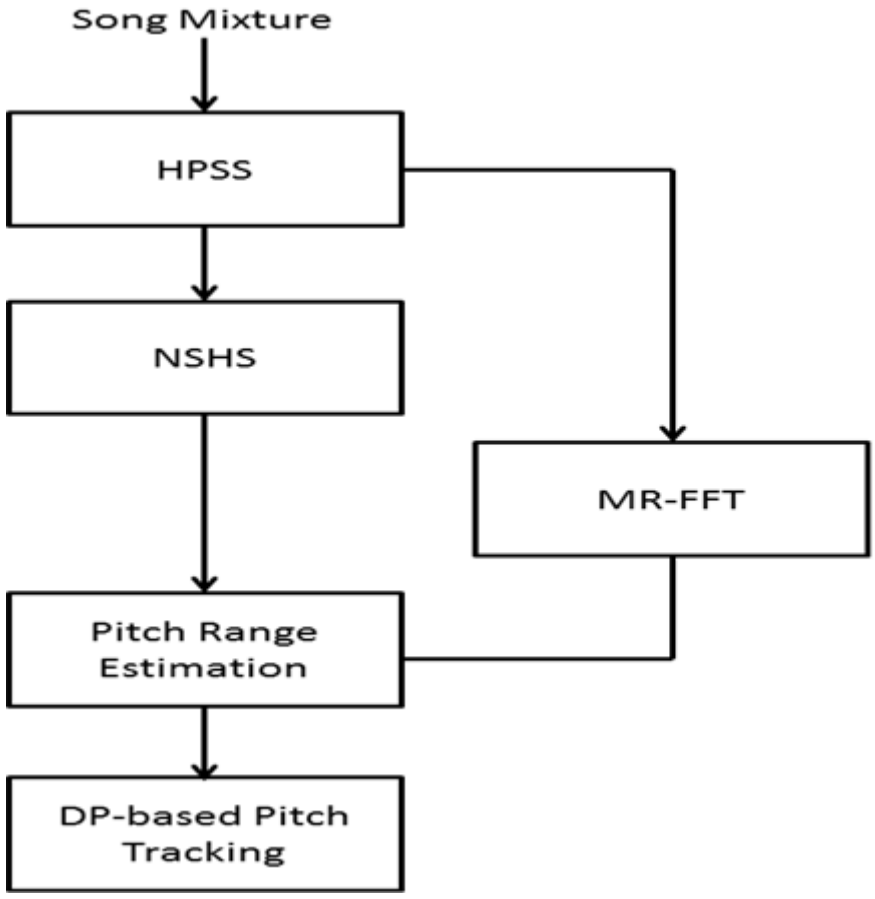


圖 3.1 Hsu 的方法流程圖[17]

Hsu 的方法可以大略分成四個部分：消除背景伴奏的 Harmonic/Percussive Sound Separation (HPSS)、將頻譜上有音高與沒音高能量差距加大的 Normalized Sub-harmonic Summation (NSHS)、推估可能人聲音高頻率範圍的 Pitch Range Estimation、及截取出音高曲線的 DP-based Pitch Tracking。3.2 節將針對每個方塊做詳細介紹。

3.2 Singing Pitch Extraction

3.2.1 Harmonic/Percussive Sound Separation (HPSS)

由於要處理的流行歌曲含有樂器伴奏，樂器伴奏有如雜訊會影響人聲音高的擷取，因此它使用 HPSS 這套方法來消除背景伴奏。

HPSS 假設訊號的功率頻譜圖可以用以下式子做拆解

$$H(m, f) + P(m, f) = W(n) \tag{3.1}$$

其中 W 是音樂訊號作 Short Time Fourier Transform (STFT)的功率頻譜，它是由兩個部分所組成，分別是 Harmonic Sound (H)與 Percussive Sound (P)，m 與 f 分別為 frame index 與 frequency bin，H 為頻譜上時間方向較平滑的訊號，如人聲、鋼琴聲等，P 為頻譜上頻率方向較平滑的訊號，如鼓聲。在上面的假設下，HPSS 可以看做是把下式化為最小值的問題:

$$J[H,P] = \omega_H \int\int \left(\frac{\partial}{\partial t} |H(t,\omega)|^{\gamma} \right)^2 dt d\omega + \omega_P \int\int \left(\frac{\partial}{\partial \omega} |P(t,\omega)|^{\gamma} \right)^2 dt d\omega \tag{3.2}$$

其中 ω_H 與 ω_P 為 constant，[8]中假設這兩個值為 1，γ 為 0.5 ，經過一連串的數學推導後可以得到以下數學式子做迭代分離出 H 與 P

$$\lambda_{t,\omega} = \left|H_{t+1,\omega}\right|^{0.5} + \left|H_{t-1,\omega}\right|^{0.5} \quad (3.3)$$

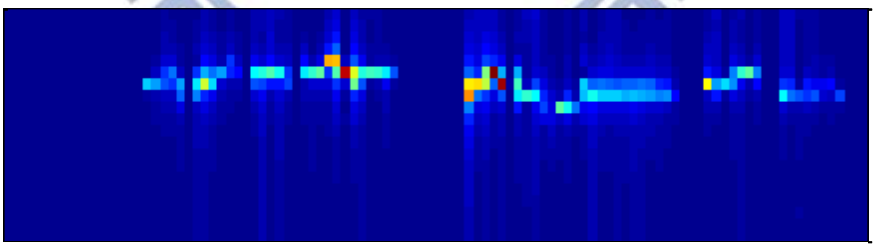
$$\mu_{t,\omega} = \left|P_{t,\omega+1}\right|^{0.5} + \left|P_{t,\omega-1}\right|^{0.5} \quad (3.4)$$

$$\left|H_{t,\omega}\right| \leftarrow \frac{\lambda_{t,\omega}^2 \left|W_{t,\omega}\right|}{\lambda_{t,\omega}^2 + \mu_{t,\omega}^2} \quad (3.5)$$

$$\left|P_{t,\omega}\right| \leftarrow \frac{\mu_{t,\omega}^2 \left|W_{t,\omega}\right|}{\lambda_{t,\omega}^2 + \mu_{t,\omega}^2} \quad (3.6)$$

之後再使用 Inverse Short Time Fourier Transform (ISTFT)轉回時域訊號，而在做 STFT 時，[8]認為一般樂器時間方向的平滑度會比人聲平滑度來的大，而人聲在時間方向的平滑度又比敲擊樂器來的大，因此可以將訊號使用不同大小的窗口做 STFT，由頻譜的差異做兩階段的處理，圖 3.2 為使用大、小窗口做 STFT 的頻譜差異比較

使用小窗口



使用大窗口

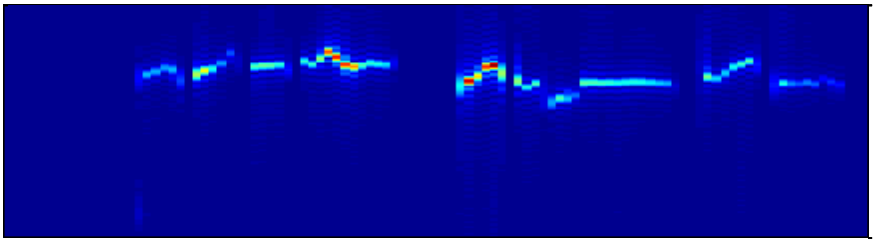


圖 3.2 使用大窗口與小窗口做 STFT 的頻譜比較圖

由圖 3.2 可以看到使用小窗口時，時間解析度會增加，頻率解析度會降低；而使用大窗口時，時間解析度會降低，頻率解析度會增加。因此，它提出第一階段使用較大的窗口(200ms)做 STFT，並使用(3.3)式到(3.6)式分離 H 與 P，此時一般樂器會被分類到 H，人聲與敲擊樂器會被分類到 P，接著第二階段使用較小的窗口(30ms)對 P 做 STFT，之後再使用(3.3)式到(3.6)式做迭代就能夠分離出人聲與敲擊樂器了，兩階段迭代的次數皆為 10 次，流程圖如下

$$W(t,\omega) \longrightarrow \left\{ H^{(1)}(t,\omega), P^{(1)}(t,\omega) \right\} \quad \text{HPSS with long window} \tag{3.7}$$

$$P^{(1)}(t,\omega) \longrightarrow \left\{ H^{(2)}(t,\omega), P^{(2)}(t,\omega) \right\} \quad \text{HPSS with short window} \tag{3.8}$$

其中 $H^{(2)}(t,\omega)$ 為我們要的人聲。圖 3.3 為純人聲音檔訊號的原始頻譜圖，理想上做完兩階段 HPSS 的 H 可以近似純人聲，下面依序呈現做完 HPSS 後的頻譜結果

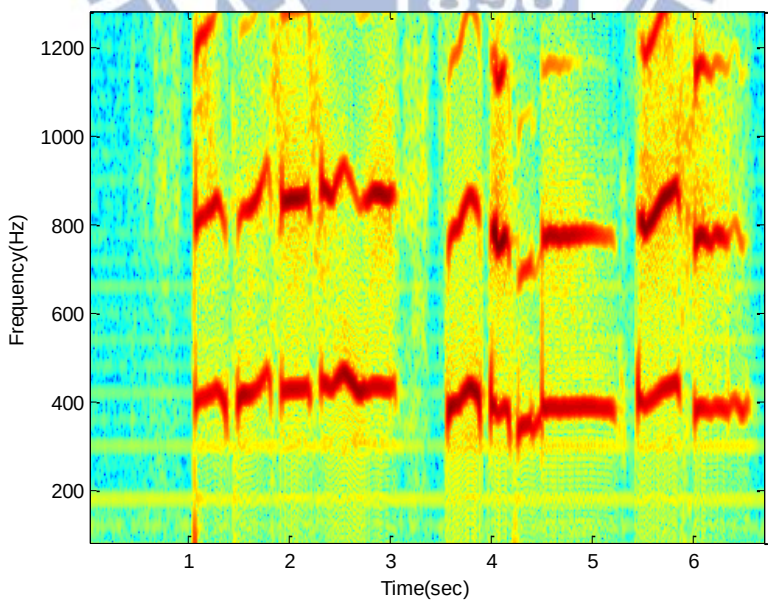


圖 3.3 純人聲音檔頻譜圖

圖 3.4 為原始流行音樂歌曲訊號頻譜圖(純人聲+背景音樂伴奏)

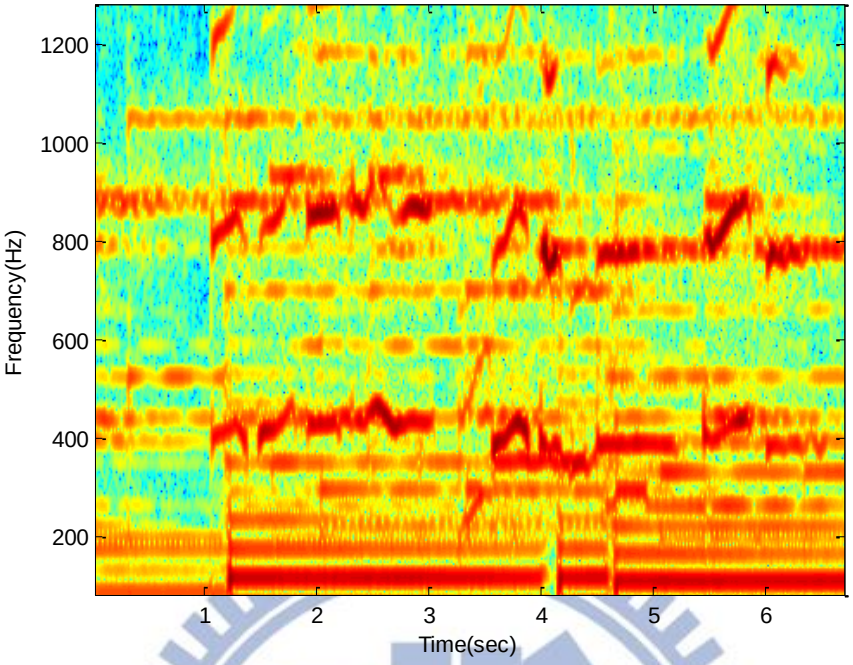


圖 3.4 一般流行音樂頻譜圖

經過 HPSS 第一階段後的 *H* 頻譜圖如圖 3.5 所示

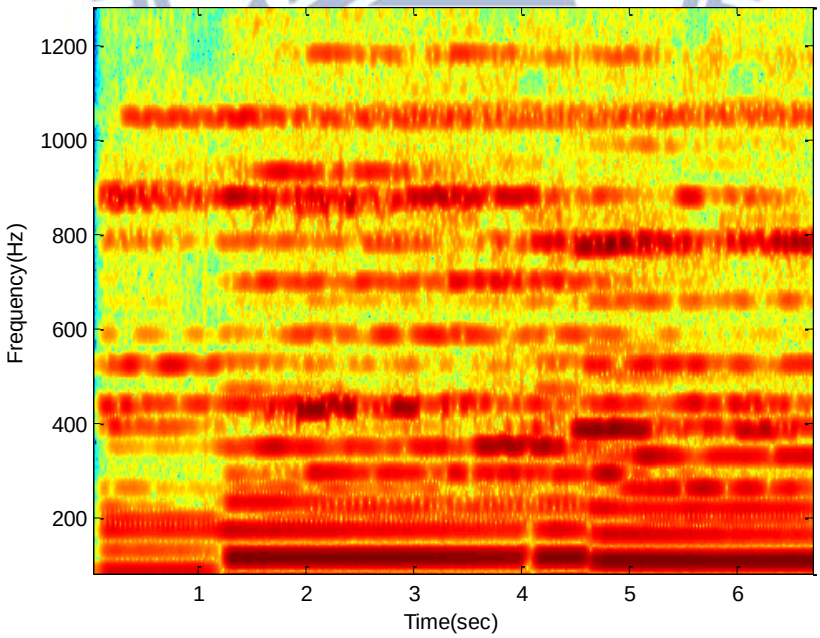


圖 3.5 HPSS 第一階段的 *H* 頻譜圖

經過 HPSS 第一階段後的 *P* 頻譜圖如圖 3.6

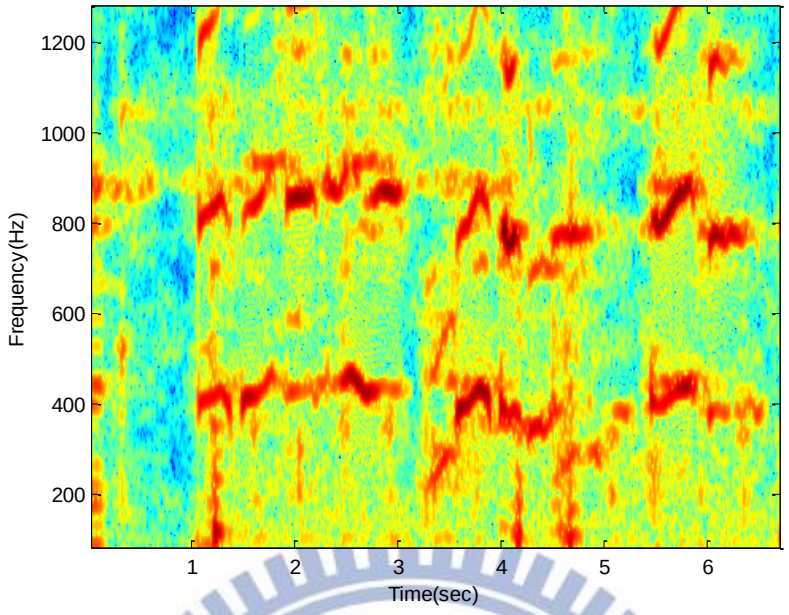


圖 3.6 HPSS 第一階段的 *P* 頻譜圖

接著對第一階段的 *P* 繼續用(3.3)式到(3.6)式做分離，圖 3.7 為 HPSS 第二階段的 *H* 頻譜圖

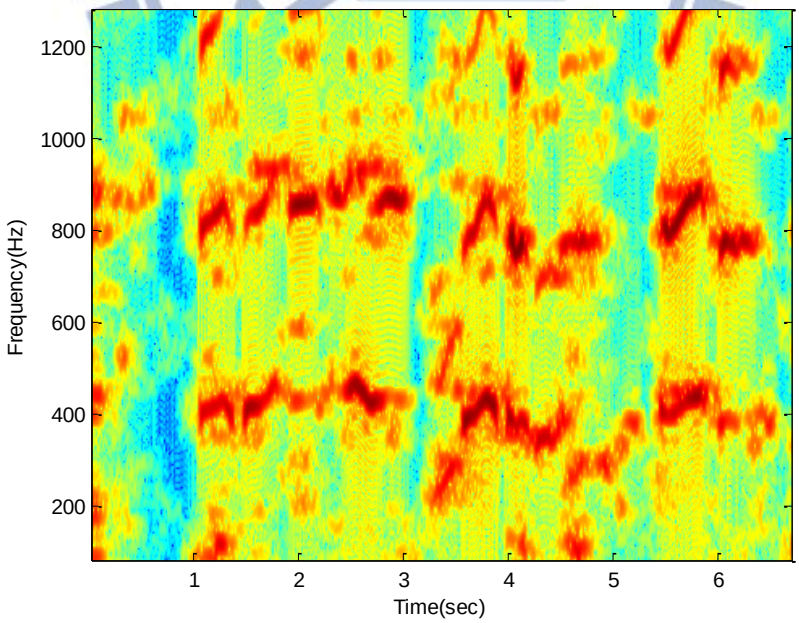


圖 3.7 HPSS 第二階段的 *H* 頻譜圖

圖 3.8 為 HPSS 第二階段的 *P* 頻譜圖

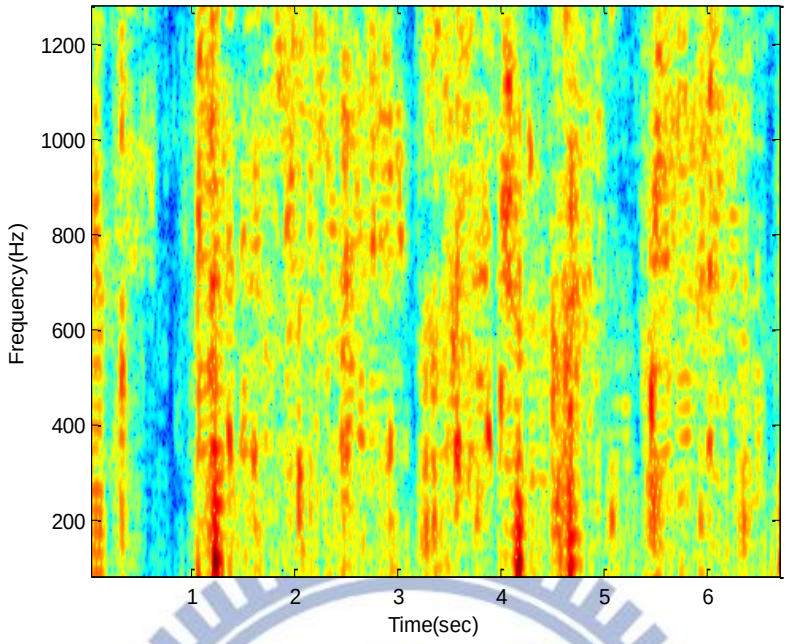


圖 3.8 HPSS 第二階段的 *P* 頻譜圖

Hsu 的方法考量到敲擊樂器本身是沒有音高的，為了不破壞音檔本身的結構，HPSS 只使用到第一階段的 *P* 去做後續處理，詳細的 HPSS 數學推導可參考[8]。

3.2.2 Normalized Sub-harmonic Summation（NSHS）

音高的計算方法很多，主要可以分為時域和頻域兩大類，一般來說，時域的方法計算速度較快，適合應用在即時的狀況下，如人在哼唱片段歌曲進入系統時，但時域的方法所計算出的音高準確度沒有頻域的方法來的好，因此頻域的方法較適合用在計算資料庫的音樂上，而 Hsu 的音高擷取方法屬於頻域的方法，最後需對頻譜上能量較大的頻率做 DP-based Pitch Tracking 找出歌唱音高曲線。為了增加有音高與沒音高的能量距離，可以用疊加諧波後的頻譜 Sub-harmonic Summation (SHS) [9]做處理，SHS 的原理是根據基頻本身的能量較高外，其倍頻諧波的能量也通常較高，因此若某一基頻與其倍頻能量總和明顯高於其他頻率時，該頻率則極有可能為基頻，SHS 方程式如下

$$H_t(f)=\sum_{n=1}^{N} h_n P_t(nf) \tag{3.9}$$

其中 $H_t(f)$ 為疊加固定諧波個數的結果，t 為 frame index，f 為 frequency bin，$P_t(*)$ 代表經過STFT後的頻譜圖，h_n 是第n個諧波權重，通常設定 $h_n = h^{n-1}, h \le 1$，這裡設定 h=0.98，N 為考慮的諧波總個數。

但是 Hsu 認為人聲在頻譜上的諧波衰減地相當慢，因此 Hsu 考慮了人聲可能存在的音高頻率的所有諧波(80Hz -1280Hz)，而非固定諧波個數，但疊加所有諧波後可以發現在低頻的地方能量會相當大，如圖 3.9 所示

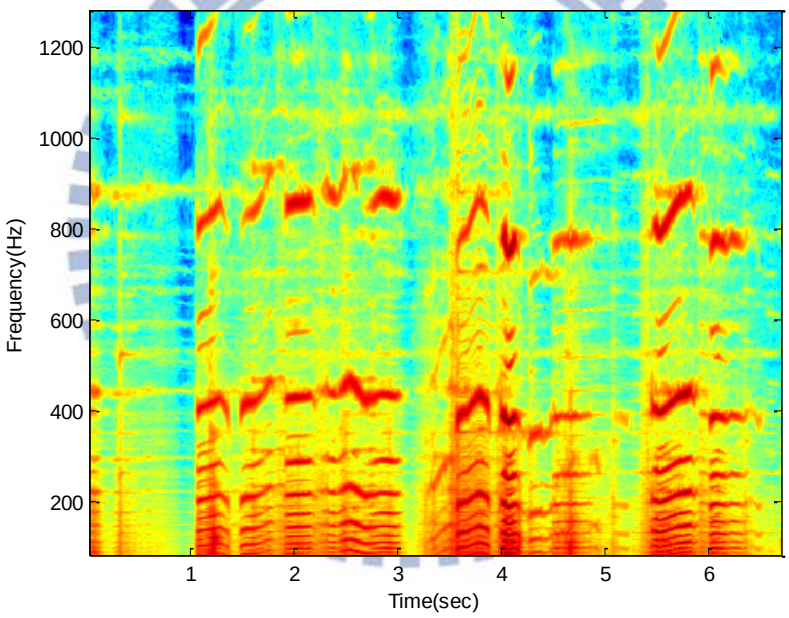


圖 3.9 未正規化的 SHS 頻譜圖

為了降低低頻的能量，改寫原先式子(3.9)做正規化，方程式如下

$$\hat{H}_t(f)=\frac{\sum_{n=1}^{N_f} h_n P(nf)}{\sum_{n=1}^{N_f} h_n} \tag{3.10}$$

其中 $N_f = floor(\frac{0.5\,fs}{f})$, fs 為取樣頻率 16k。圖 3.10 為使用 HPSS 第一階段 P 所做出的 NSHS 頻譜圖

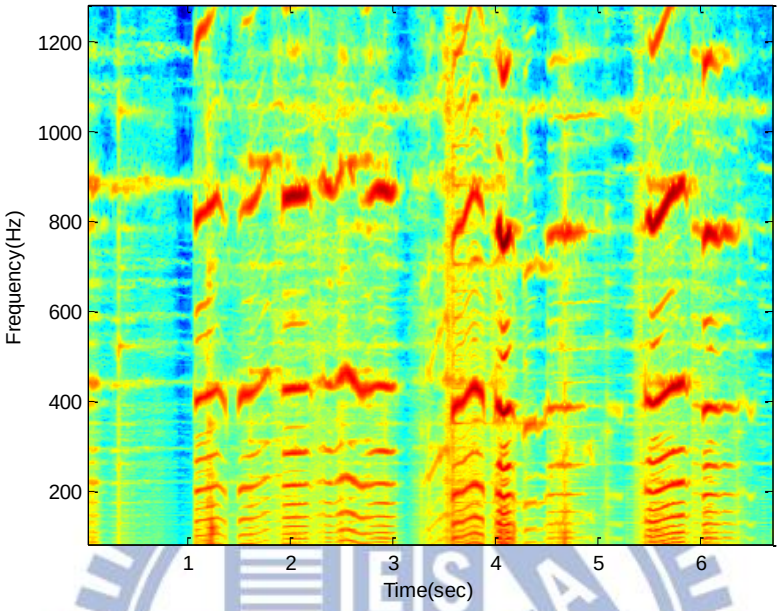


圖 3.10 NSHS 頻譜圖

可以看到做完正規劃的 NSHS 在低頻處明顯較 SHS 來得清晰。

3.2.3 Pitch Range Estimation

為了消除頻譜上的諧波，找出歌唱音高，它藉由 Dressler 在[10]提出的 Multi-resolution Fast Fourier Transform (MR-FFT)找出人聲音高頻率可能的範圍，MR-FFT 的相關理論說明如下。

MR-FFT 可以擷取出音樂訊號的正弦成份，並消除頻譜上來自於雜訊等不可靠的峰值，概念上是利用不同的時間-頻率解析度來重新合成頻譜，例如對一首取樣率為 44.1 *k*Hz 的音檔做 4 次 FFT，每次 FFT 的窗口都固定為 8192 點，第一次針對低頻的部分取大窗口 2048 點做 FFT，不夠的做補零的動作(zero padding)，小於一門檻值的峰值將不被考慮，第二次考慮中頻的部分，做 1024 點 FFT，不夠的做補零的動作，小於一門檻

值的峰值也不考慮，這樣依此類推做到第四次高頻的部分，最後把做出的結果疊加起來，結果如圖 3.11 所示。

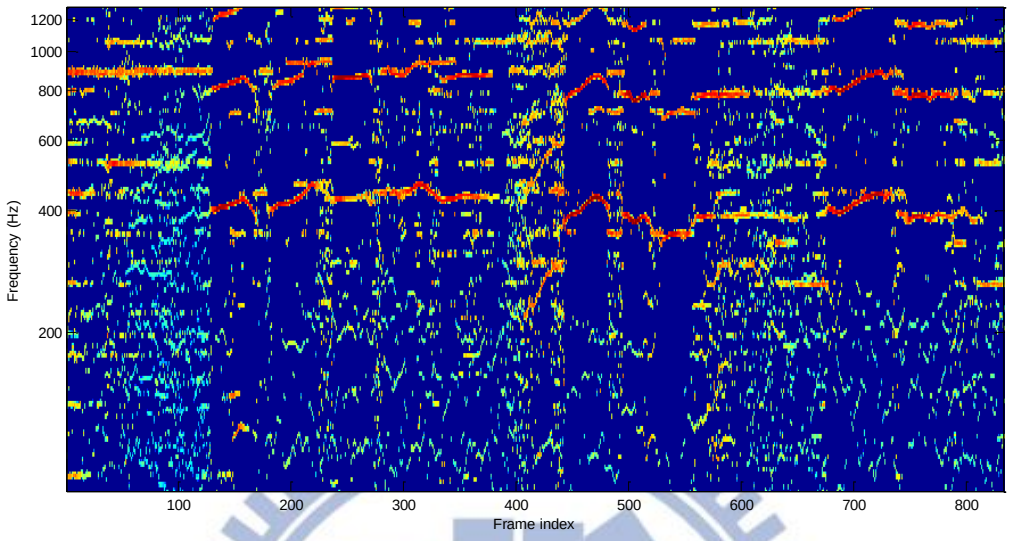


圖 3.11 MR-FFT 頻譜圖

接著觀察每個音框的瞬間頻率，把頻率相近的峰值(只差距 0.2 個半音以內的)選能量最大的留下，結果如圖 3.12，可以看到它比圖 3.11 更為清晰

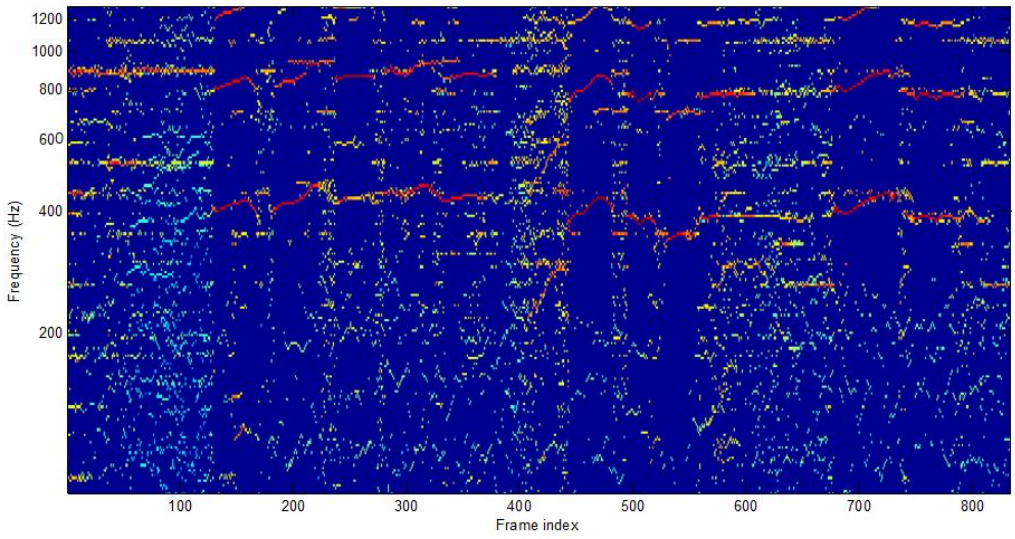


圖 3.12 刪減峰值後的 MR-FFT 頻譜圖

這裡假設基頻為最低頻率，因此找出每個音框中最小的頻率值，刪減其諧波，其結果如

圖 3.13 所示，已經可以看到約略的人聲音高曲線。

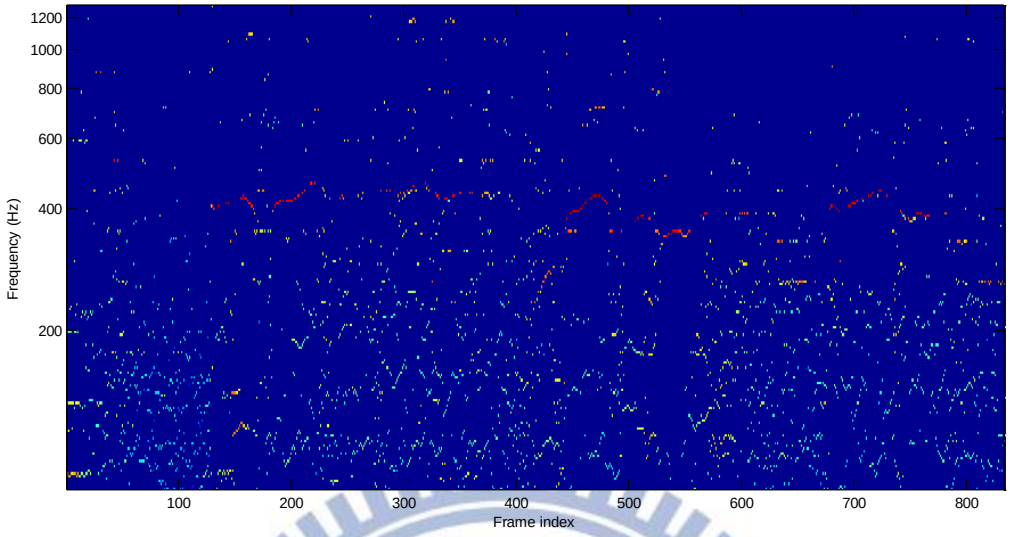


圖 3.13 刪除諧波後的 MR-FFT 頻譜圖

為了找出人聲音高的頻率範圍，它以刪減諧波後的頻譜 $x[m, f]$ 為基礎建立多個時間頻率區塊（T-F Blocks），每個區塊在時間與頻率方向的交疊都是 1/2，並以下面公式計算區間裡的能量大小，T-F Blocks 能量公式如下

$$b(T,F) = \sum_{m=0}^{M_T-1} \max_{f\in[0,M_F-1]} x[m, TL_T + f, FL_F] \tag{3.11}$$

其中 $T = 0,1,...P-1$ and $F = 0,1,...Q-1$。上式中 T 與 F 代表了 T-F Block 中在時間軸與頻率軸上的索引值，P 與 Q 為 T-F Block 中時間與頻率的範圍，M_F 與 M_T 為一個 block 所包含的 frequency bin 與 frame index 數目，L_F 與 L_T 代表在 $x[m, f]$ 中頻率與時間上平移的大小，這裡 M_T 設定成 188 個 frame index，M_F 設定成 7.75 個半音值，L_T 設定成 94 個 frame index，而 L_F 設定成 3.875 個半音值，做完 T-F block 後得出的能量結果如圖 3.14 所示。

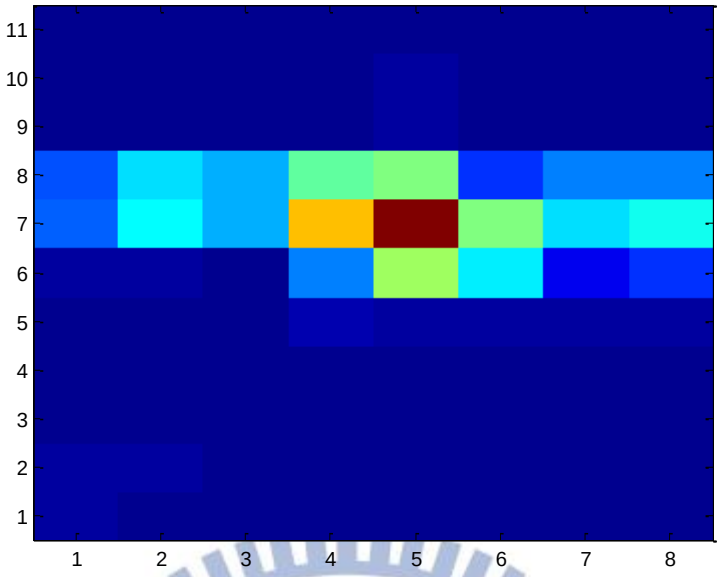


圖 3.14 T-F block 能量分佈圖

根據計算出的能量分佈圖可以用 DP 的概念計算其動態規劃能量分數，它考慮到目前所在音框中的頻率索引可能來自前一個音框中的任意頻率索引，但由於音高曲線為一個平滑的曲線，因此需加上一個 penalty factor 控制平滑度來算出動態規劃能量分數，最後再逆向回朔找出最佳路徑。式(3.12)為動態規劃能量分數的計算公式，圖 3.15 為 DP 的示意圖，符號藍色★為目前考慮的位置，它可能來自前一個音框的 5 個方向，斜率最大的方向在式(3.12)中第二項會較大(黃色部分)，而斜率為 0 的方向第二項會為 0。

$$score(T,F)=\max_{i\in[0,Q-1]}\{score(T-1,F-i)+b(T,F)+\theta\times|i|\} \tag{3.12}$$

其中，$score(T,1)=b(T,1),T=0,1,...P-1$，$\theta=\frac{1}{2}\left(\frac{\sum_{T=0}^{P-1}\max_{F\in[0,Q-1]}b(T,F)}{P}\right)$

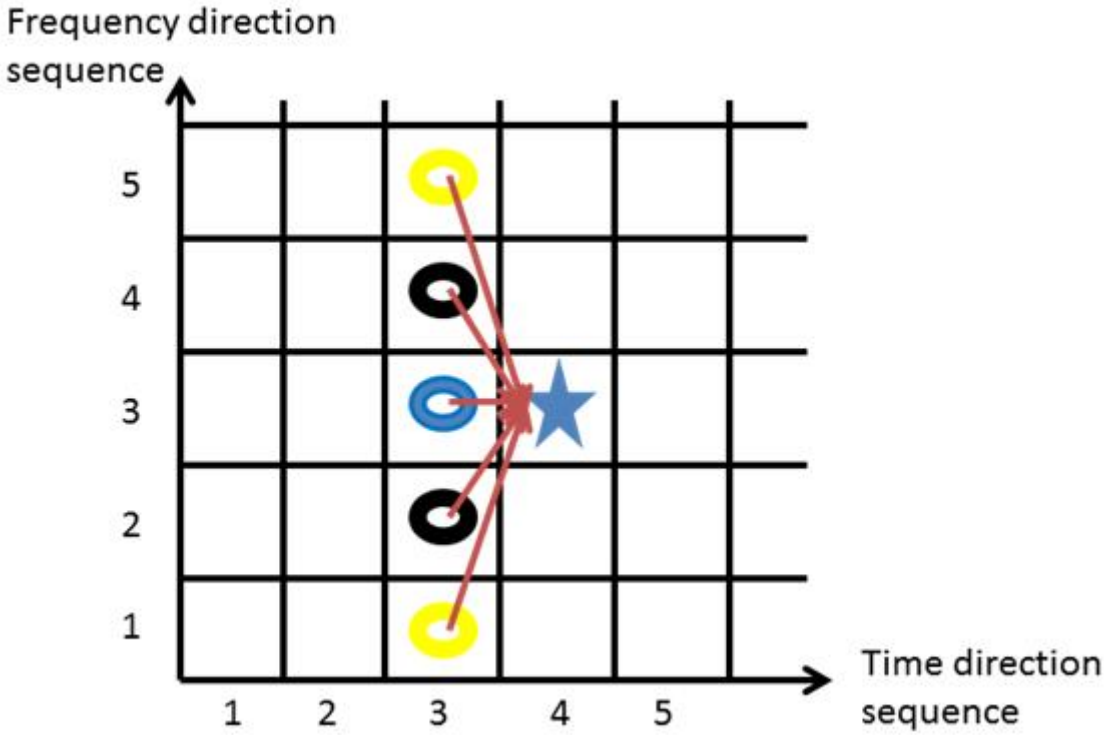


圖 3.15 DP 示意圖

圖 3.16 為在 T-F block 使用 DP 的結果。

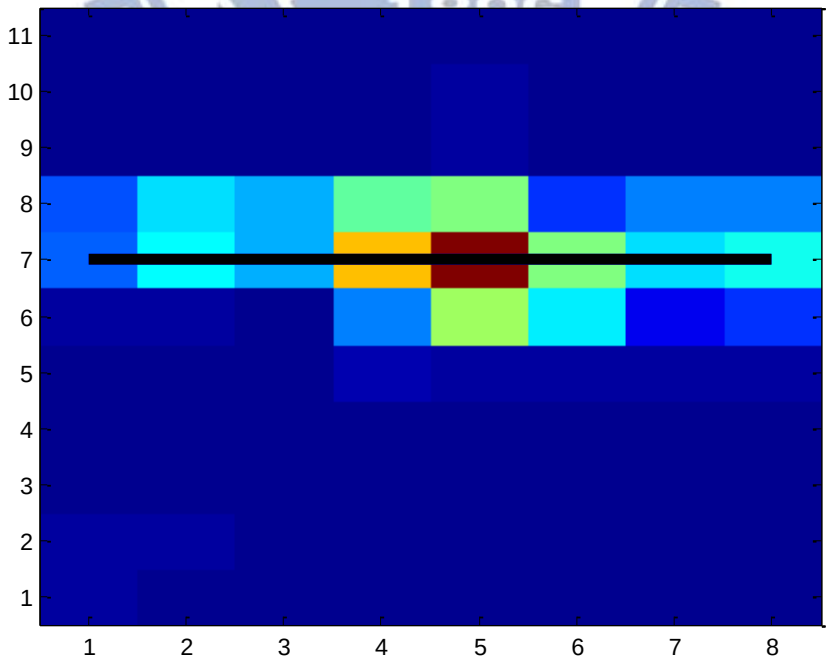


圖 3.16 動態規劃找出的最佳路徑

接著它以能量最強的 block 為基礎，計算其頻率方向相鄰 block 的相關性，若 $b(T,F-1)/b(T,F+1)$ 小於 0.8 表示人聲音高較可能在高頻處，若 $b(T,F+1)/b(T,F-1)$ 小於 0.8 則表示其人聲音高較可能出現在低頻處，因此使用下面公式可以得出能量在哪個範圍會較強，也就是人聲音高頻率較可能出現的地方

$$\begin{cases} [f_{T,F}^{lower}, f_{T,F}^{upper}+\Omega] & \text{if } b(T,F-1)/b(T,F+1)<0.8 \\ [f_{T,F}^{lower}-\Omega, f_{T,F}^{upper}] & \text{if } b(T,F+1)/b(T,F-1)<0.8 \\ [f_{T,F}^{lower}-\frac{\Omega}{2}, f_{T,F}^{upper}+\frac{\Omega}{2}] & \text{otherwise} \end{cases} \quad (3.13)$$

其中 $\Omega=4$ semitones，而 $f_{T,F}^{lower}$ 與 $f_{T,F}^{upper}$ 為黑線所在 block 的半音值，經過(3.13)式後即可知道人聲音高頻率範圍變化如圖 3.17 所示

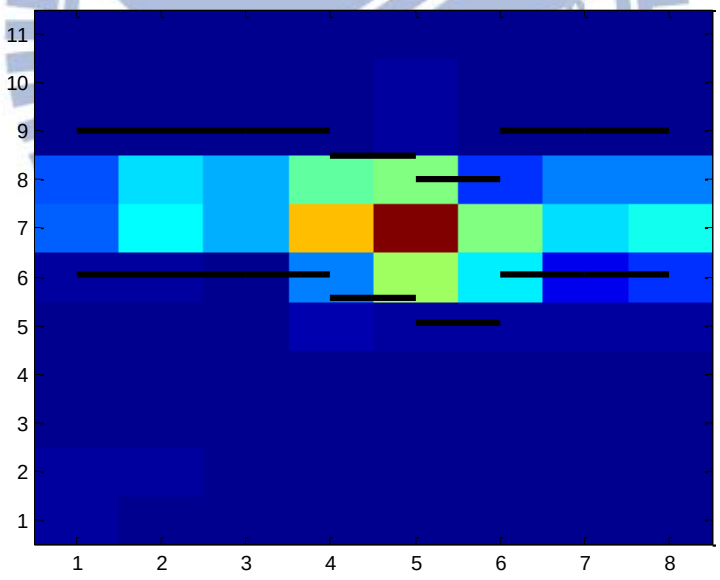


圖 3.17 T-F block 上人聲變化趨勢

圖 3.18 為用以上方法所估出的人聲音高頻率範圍在 NSHS 上的結果。

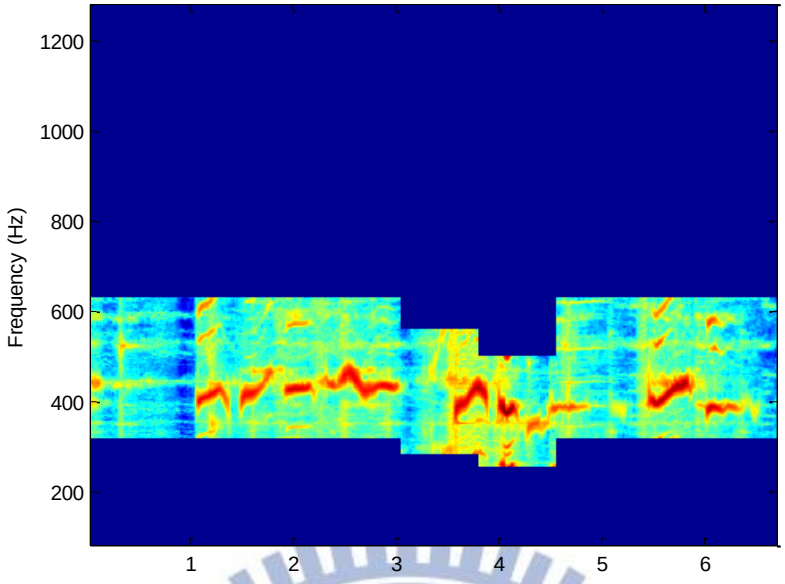


圖 3.18 所估計出的人聲音高頻率範圍

3.2.4 DP-based Pitch Tracking

利用 3.2.3 節的方法確定歌聲可能存在的頻率範圍後，再使用 Dynamic Programming 找出最後的音高曲線，實驗設定懲罰因子θ為 2，考慮的人聲音高頻率範圍為 40 semitones (82Hz)到 80 semitones (830Hz)，為了加快 DP 的速度，它使用式(3.14)找出每一個半音值在頻譜上最接近的頻率刻度，並擴展其在 NSHS 頻譜中的頻率範圍$\pm\tau$來找出其間能量最大的峰值與其所對應到的頻率，τ的計算方式如式(3.15)

$$freq = 440 \times 2^{\frac{semitone-69}{12}} \tag{3.14}$$

$$\tau(i) = \left\lceil \frac{440 \times 2^{\frac{(i+1)-69}{12}} - 440 \times 2^{\frac{(i)-69}{12}}}{2} \right\rceil \tag{3.15}$$

其中i表示目前半音值，$\lceil\ \rceil$表示取出轉換後不小於其值的整數中最小的值。圖 3.19 為降低解析度後的峰值能量分佈，根據圖 3.19 可以使用 DP 算出動態規劃能量分數，最後再逆向回朔找出最佳路徑，並根據這個最佳路徑找出原先峰值所對應的頻率值，圖 3.20 中藍線為 Hsu 的方法所求出的音高曲線，其方法的詳細推導過程可以參考[3]與[17]。

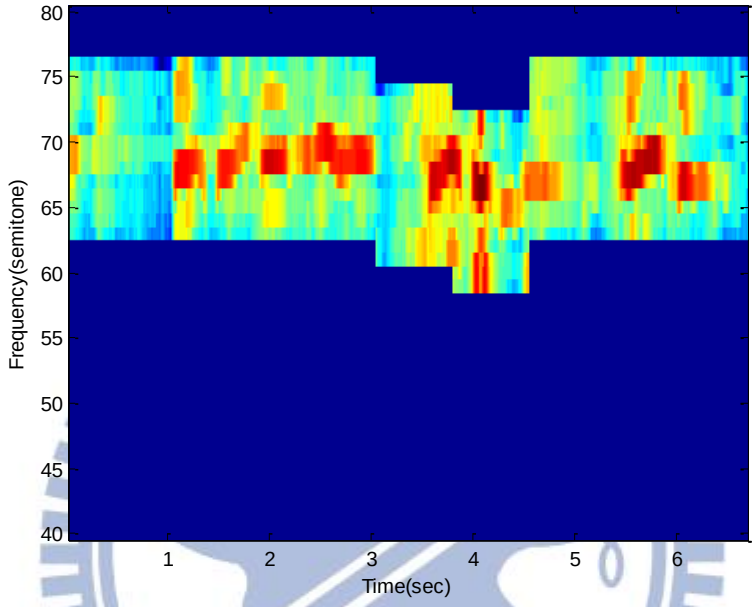


圖 3.19 NSHS 峰值能量圖

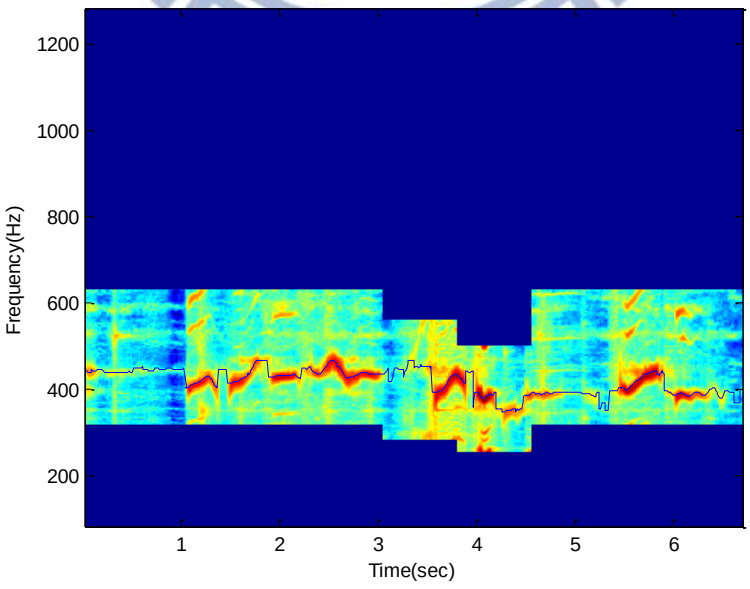


圖 3.20 Hsu 的方法所估計的音高曲線圖

第四章 使用單音 MIDI 改善音高曲線

Hsu 的方法已經算是準確度相當高的方法，然而它還是存在某些缺點，以圖 4.1 來看，縱軸為半音值，橫軸為 Frame index，紅色的線為人工標記的音高答案，而藍色的線為 Hsu 的方法所算出的音高曲線，可以看到歌聲邊緣與歌聲頻率差距大時容易有錯誤發生，原因為 Hsu 的方法只考慮到 Raw Pitch，而伴奏樂器的平滑度是我們無法掌控的，與人聲銜接時可能會有一段音高上的落差，因此本研究將加入對齊後的單音 MIDI 來縮小人聲音高頻率範圍擷取更準確的音高，並判斷人聲段與非人聲段，用以在後續使用基於動態規劃的方法擷取音高曲線時，可以忽略無人聲部分的平滑度，因而可以降低人聲端點偵測可能產生的錯誤。

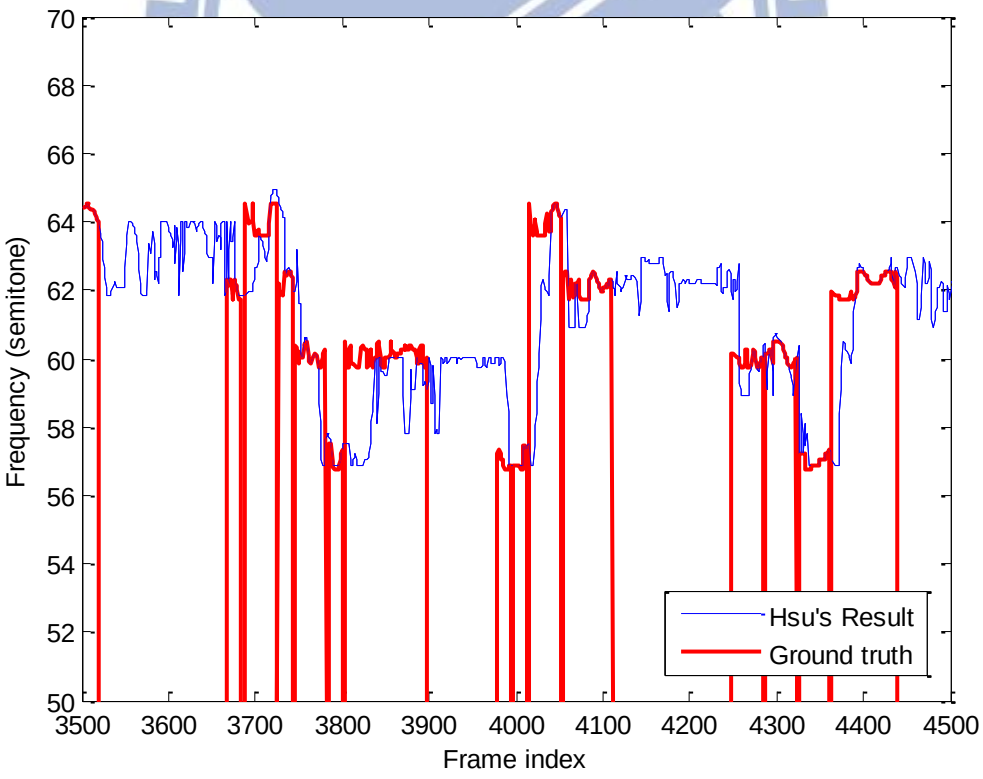


圖 4.1 Hsu 的方法所求出的音高曲線與人工標記音高答案

4.1 使用單音 MIDI 改善音高偵測曲線

首先，系統會使用兩種類型的音檔，分別為單音 MIDI 檔與其對應的流行歌曲，單音 MIDI 的調(Key)將先調成與 Hsu 的方法所求出的音高曲線平均值相同，並使用[12]所提供的 MIDI 合成器合成領航音檔(pilot audio)，而在流行歌曲方面，由於這裡的流行歌曲含有背景伴奏，並非純人聲音樂(singing voice)，因此將沿用第三章所介紹的 HPSS 來消除樂器伴奏，之後對 HPSS 處理後的流行歌曲與領航音檔的特徵值做動態時軸校對，找出它們的時間對應關係，而經過觀察發現會有不自然的音符發生，因此將使用一套後處理的技巧來做對齊時間的微調修正，並再一次修改單音 MIDI 的調，之後利用對齊好的單音 MIDI，重新估計人聲音高頻率範圍，並判斷人聲段與非人聲段，最後再使用 DP 求出音高曲線，圖 4.2 為使用對齊單音 MIDI 來改善音高偵測曲線的流程圖，以下各節對此流程圖的各方塊做說明。

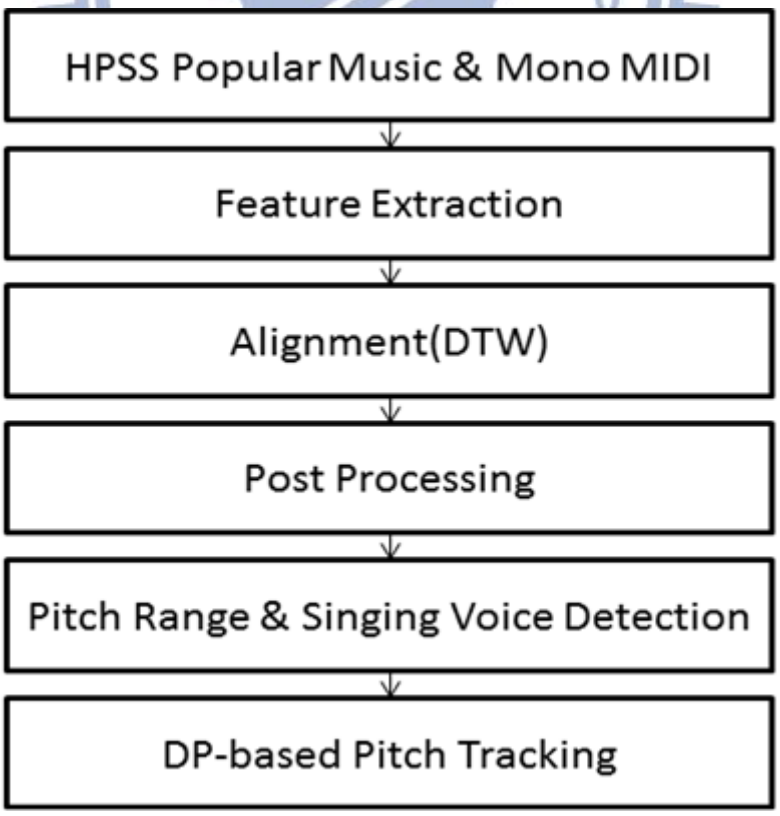


圖 4.2 使用單音 MIDI 改善音高曲線流程圖

4.2 Aligning Popular Music with Mono MIDI

4.2.1 HPSS Popular Music & Mono MIDI

MIDI_888 資料庫中的流行歌曲是使用[7]的方法來把 YouTube 上非 MV 版本的歌曲下載下來，並統一為單聲道，取樣率為 16*k*Hz，解析度為 16-bit 的 wav 檔，由於參考論文[11]所使用的是純人聲音檔，因此將使用前一章所提到的 HPSS 來達到近似純人聲的效果，為了消除敲擊樂器的影響，將會使用到 HPSS 第二階段的 H 來做後續處理，而針對單音 MIDI，會先把其調修改成與 Hsu 的方法所求出的音高曲線平均值相同，然而考慮的是整首歌曲的調而非只考慮人聲段部分，因此調整後的調可能會有略低或略高的情況發生，如圖 4.3 所示

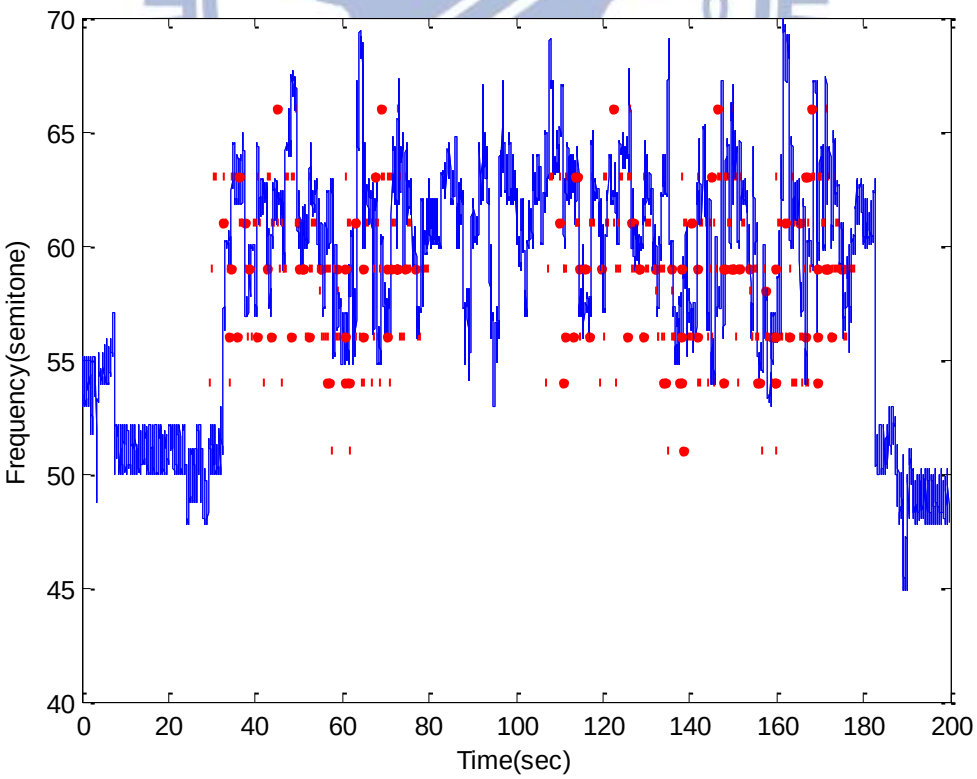


圖 4.3 第一次調 key 示意圖

圖中藍線為 Hsu 的方法所求出的音高曲線，而紅線為尚未對齊的 MIDI 音符，從圖中可以看到 MIDI 音符有略低的趨勢，針對此問題之後將利用對齊後的單音 MIDI 來解決，而 MIDI 轉 wav 的技術是利用[12]重新合成領航音檔，其為單聲道，取樣率為 8*k*Hz，解析度為 16-bit 的音檔。

4.2.2 Feature Extraction

HPSS 理論上第一階段的 *P* 只剩人聲與敲擊樂器，而第二階段的 *H* 只剩人聲，由於上一章的重點是音高擷取，而敲擊樂器沒有音高，為了不破壞音檔品質，Hsu 的方法只用到第一階段的 *P*，而針對 MIDI 與流行歌曲做時間對齊這動作，為了消除敲擊樂器的影響，將會使用 HPSS 第二階段的 *H* 來做特徵參數抽取，HPSS 第二階段的 *H* 音檔取樣率為 16*k*Hz，為了增加程式執行的效率，將降低取樣率為 8*k*Hz，之後分別對 HPSS 第二階段的 *H* 音檔與單音 MIDI 領航音檔建立頻譜，實驗設定每個音框的大小為 1024/8000 =0.128s，重疊的大小為 768/8000=0.096s，而為了增加頻率解析度，將以 8192 點來做 STFT。

這裡抽取的特徵參數為[13]所提出的，由於本論文要對齊的是歌聲，其基頻一般而言會比較接近 MIDI 音符，而非任意頻率，因此對原本的頻譜做了頻率刻度轉換，使用以下公式把頻率單位轉換到 MIDI 單位(semitone)

$$U(j)=\left\lfloor 12\times\log_2\left(\frac{F(j)}{440}\right)+69\right\rfloor \tag{4.1}$$

其中 *F*(*j*)為原始頻率，⌊ ⌋表示取出轉換後小於或等於其值的最大整數。圖 4.4 為鋼琴上各個琴鍵所對應到的 MIDI 音符與頻率示意圖

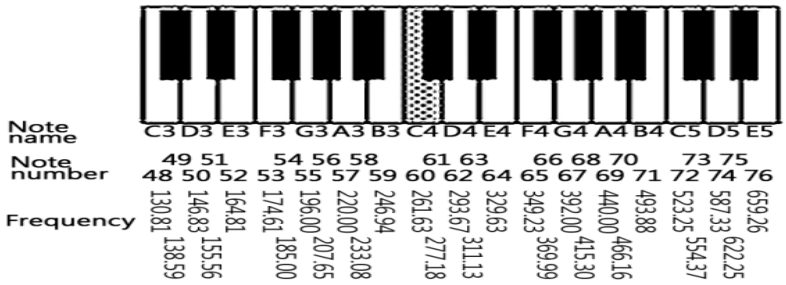


圖 4.4 琴鍵音高與頻率對照圖[18]

在頻率轉換到 MIDI 單位後，令 $X_{t,j}$ 為頻譜上第 t 個音框訊號在第 j 個 FFT 頻率刻度上的振幅，其中 $1<j<J$，而 J 為頻率刻度總數，找出其中能量最大值來代表所在的 MIDI 刻度能量，其數學式子表示為

$$Y_{t,n} = \max_{\forall j, U(j)=e_n} X_{t,j} \tag{4.2}$$

其中 e_n 表示對應的 MIDI 音符，$Y_{t,n}$ 為其能量。圖 4.5 為流行音樂原始頻譜與經過頻率刻度轉換後的頻譜的差別比較。

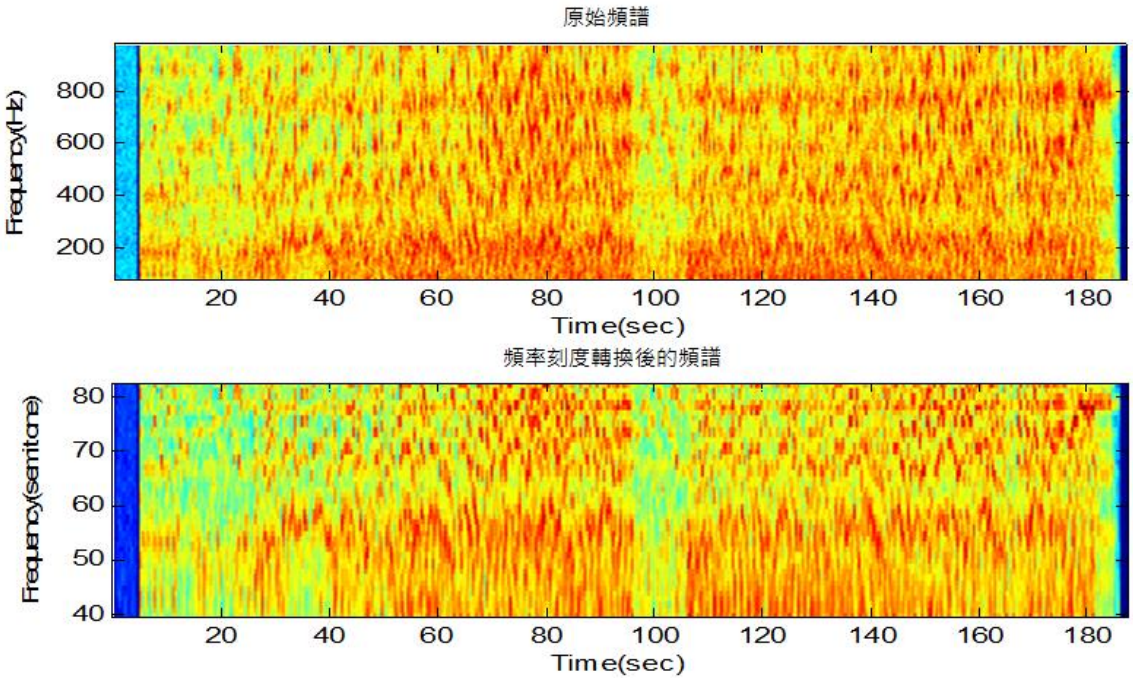


圖 4.5 流行音樂原始頻譜與經過頻率刻度轉換後的頻譜比較圖

在兩信號頻率刻度轉換後的頻譜求出後，為了降低其相異性，將用以下公式做正規化

$$F_n = (F_r - \mu)\ \sigma \tag{4.3}$$

其中 F_r 為原本特徵參數，F_n 為做完正規化後的特徵參數，μ 為特徵參數平均值，σ 為特徵參數標準差，做完正規化後的流行音檔與領航音檔的頻率刻度轉換頻譜分別如圖 4.6 所示

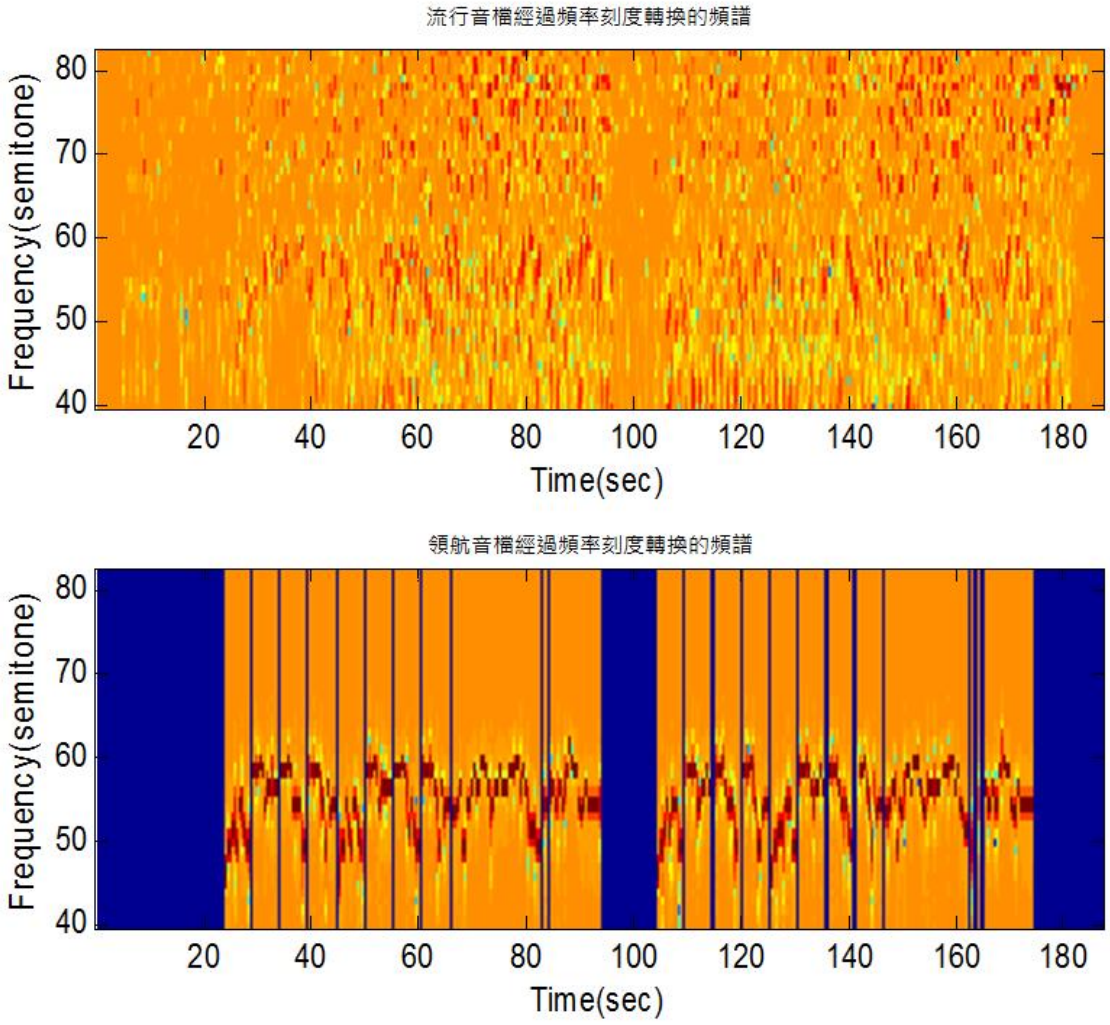


圖 4.6 流行音檔與領航音檔的頻率刻度轉換頻譜比較圖

4.2.3 Alignment by DTW

這裡所使用的對齊方法為 DTW，中文稱為動態時軸校對，DTW 是一套根基於動態規劃的方法，它可以找出兩個時間長度不同的信號彼此的對應關係，也是一種計算相似度的演算法。

假設兩個訊號時間長度不同，DTW 會分別計算出兩個訊號的特徵參數，並使用下式計算相似矩陣

$$SM(i,j) = \frac{spec_{syn}(i)^T \cdot spec_{raw}(j)}{\|spec_{syn}(i)\| \cdot \|spec_{raw}(j)\|}, \text{ for } 0 \leq i < M, 0 \leq j < N \quad (4.4)$$

其中 $spec_{syn}(i)$ 為領航音檔的特徵參數，$spec_{raw}(j)$ 為流行歌曲的特徵參數，符號$\| \|$為一般歐幾里得範數，計算出的相似矩陣如圖 4.7 所示。

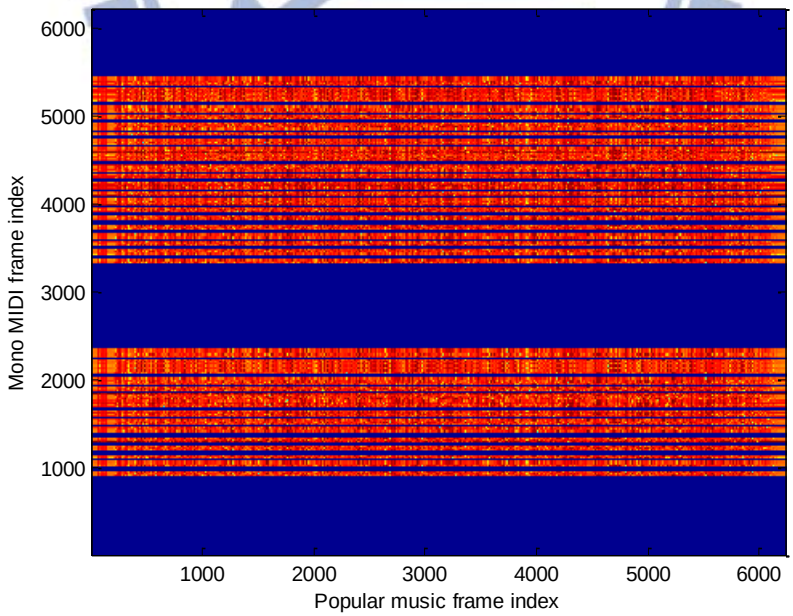


圖 4.7 相似矩陣示意圖

此相似矩陣的維度為 $M \times N$，而 DTW 的設計為找出最低代價路徑(lowest cost path)，因此

將改由矩陣 D=1-SM 做 DTW，DTW 將由 D 的原點到(M-1,N-1)找出最佳路徑，而 DTW 一般公式如下：

$$D'(i,j)=\min\begin{pmatrix}D(i,j-1)\\D(i-1,j-1)\\D(i-1,j)\end{pmatrix}+\text{distance}(i,j)$$

$$\text{distance}(i,j)=D(i,j)$$

$$D(i,1)=\infty, i=2,...,M \tag{4.5}$$

$$D'(1,1)=\text{distance}(1,1)$$

$$D(1,j)=\infty, i=2,...,N$$

其中 $D'(i,j)$ 為 DTW table 中某一點的累積分數，它的計算方式為目前的點 $D(i,j)$ 可能來自 3 個不同的方向，找出與目前的點差距最小的方向並記錄到 $D'(i,j)$。在計算完動態規劃各點的累積分數後，$D'(M,N)$ 即是兩首歌曲的距離。而本論文要對齊的兩首歌曲，存在著很大的相似性，因此做了速度上的限制，改寫後的 DTW 公式如下

$$D'(i,j)=\min\begin{pmatrix}D(i-1,j-2)\\D(i-1,j-1)\\D(i-2,j-1)\end{pmatrix}+\text{distance}(i,j)$$

$$\text{distance}(i,j)=D(i,j)$$

$$D(i,1)=\infty, i=2,...,M \tag{4.6}$$

$$D'(1,1)=\text{distance}(1,1)$$

$$D(1,j)=\infty, i=2,...,N$$

此公式表示兩首歌曲的速度差距最快與最慢都限制到 2 倍以內，其動態規劃累積分數計算方法的示意圖如圖 4.8。

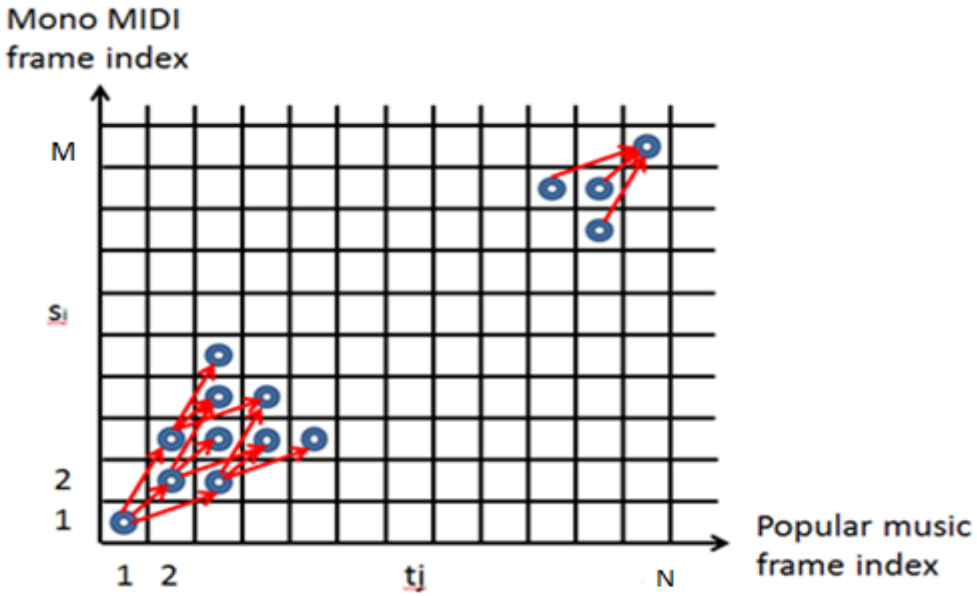


圖 4.8　DTW 示意圖

圖中每一個點都是由 3 個點計算而來，而每一個點需要被 3 個點所參考，最後只要利用右上角的 $D'(M,N)$ 逆向回朔，即可找出兩首歌曲的最佳路徑，如圖 4.9 黑線所示

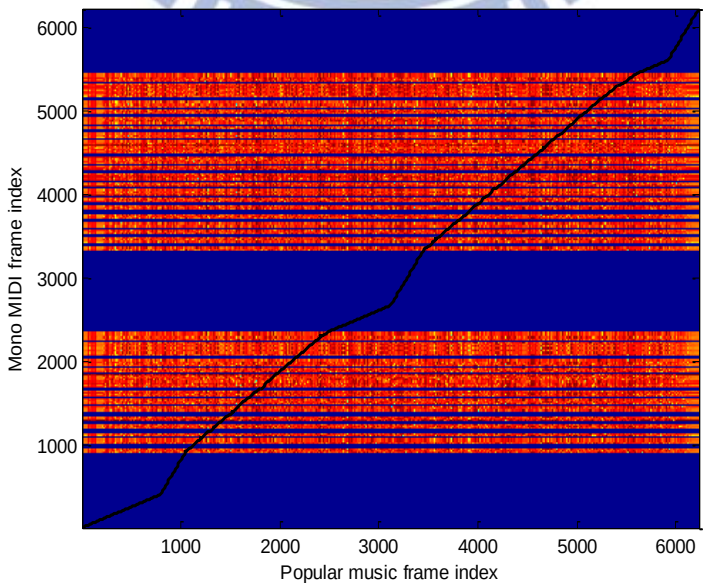


圖 4.9 DTW 後的結果

之後將根據這條黑線來找出流行音檔中 MIDI 音符所對應的開始時間與結束時間位置。

4.2.4 Post Processing

在考慮到要做對齊的單音 MIDI 檔，其主歌(verse)、副歌(chorus)的起始位置時間可能是不準的，因此需要做主歌、副歌起始時間的對齊工作。經過前一小節的 DTW 處理可以得出每個音符對應到流行歌曲的時間位置，透過觀察發現，對齊好後的音符會發生不自然的現象，有些音符會過長或過短，有些音符甚至會消失，如圖 4.10 所示，最上面的數字表示歌曲的時間位置，上圖為原始單音 MIDI，下圖為對齊完後的單音 MIDI

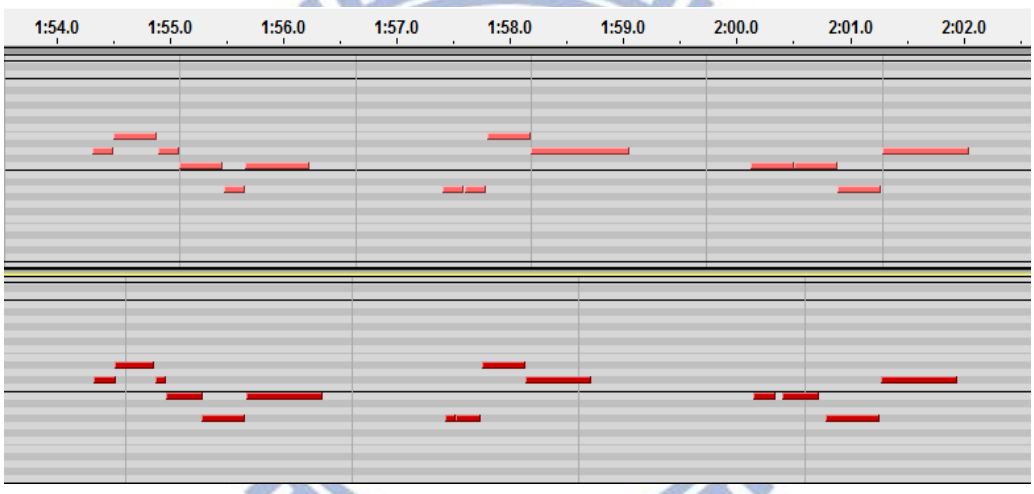


圖 4.10 原始單音 MIDI 與對齊完後的單音 MIDI

而根據 MIDI 資料庫的特性，將以 segment 為單位做處理，判斷的方法係依據原始單音 MIDI 中每個音符的結尾時間與下個音符的起頭音時間差距的平均值(μ)與標準差(σ)，並根據高斯分佈中標準偏差的特性設計一門檻值 $\mu+3\sigma$，並設定最小門檻值不小於 2 秒，若目前音符與下個音符差距大於這個門檻值，代表目前音符與下個音符差距過大，那目前所在音符即算是一個段落截止音符，圖 4.11 為“小薇”這首歌曲的 segment 判斷結果示意圖

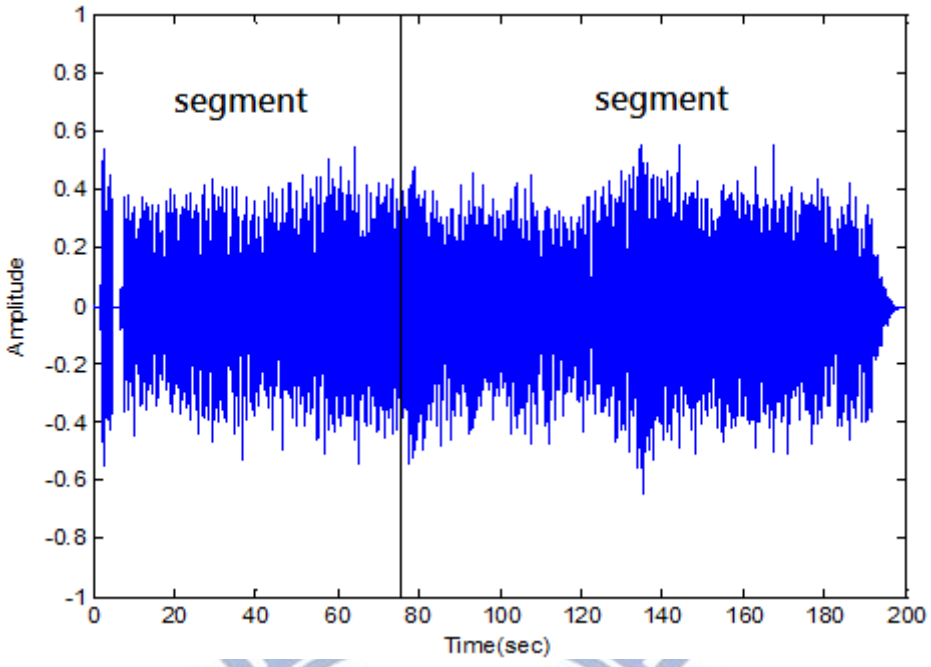


圖 4.11 segment 判斷示意圖

圖中黑線為未對齊前的段落截止音符時間位置，依照這個方法判定，可以看到“小薇”這首歌曲被分成兩個部分，而有一些歌曲判斷出的 segment 會有太短的現象，實驗中設定小於 25 個音符即為太短，將把此 segment 併入下一個 segment 做處理。

原始的 MIDI 資料庫中的單音 MIDI 音檔是由人工看著樂譜彈奏所製作出來的，理想上其每個音符的時間相對位置應有相當高的準確性，因此各自對原始的單音 MIDI 與對齊後的單音 MIDI 擷取每個音符的起頭音時間建立時間陣列，分別以符號 T_r 與 T_p 表示，之後計算其各自時間陣列裡倆倆元素的差值，也就是進行差分運算，因為差分運算是計算陣列裡後面的元素減去其前面一個元素的差值，所以差分後的結果會比原來時間陣列的元素個數少 1，其各自差分的結果分別以符號 ΔT_r 和 ΔT_p 表示，其元素的意義為音符與下一個音符的相對時間關係，因為原始的單音 MIDI 其時間相對位置假設是正確的，因此假設其起頭音時間位置的差分結果也相對正確，計算式子(4.7)將可以知道不自然音符的位置

$$\text{Note\_index} = \Delta T_r - \Delta T_p \tag{4.7}$$

圖 4.12 為 Note_index 的示意圖，其中黑線為 segment 的判斷邊界

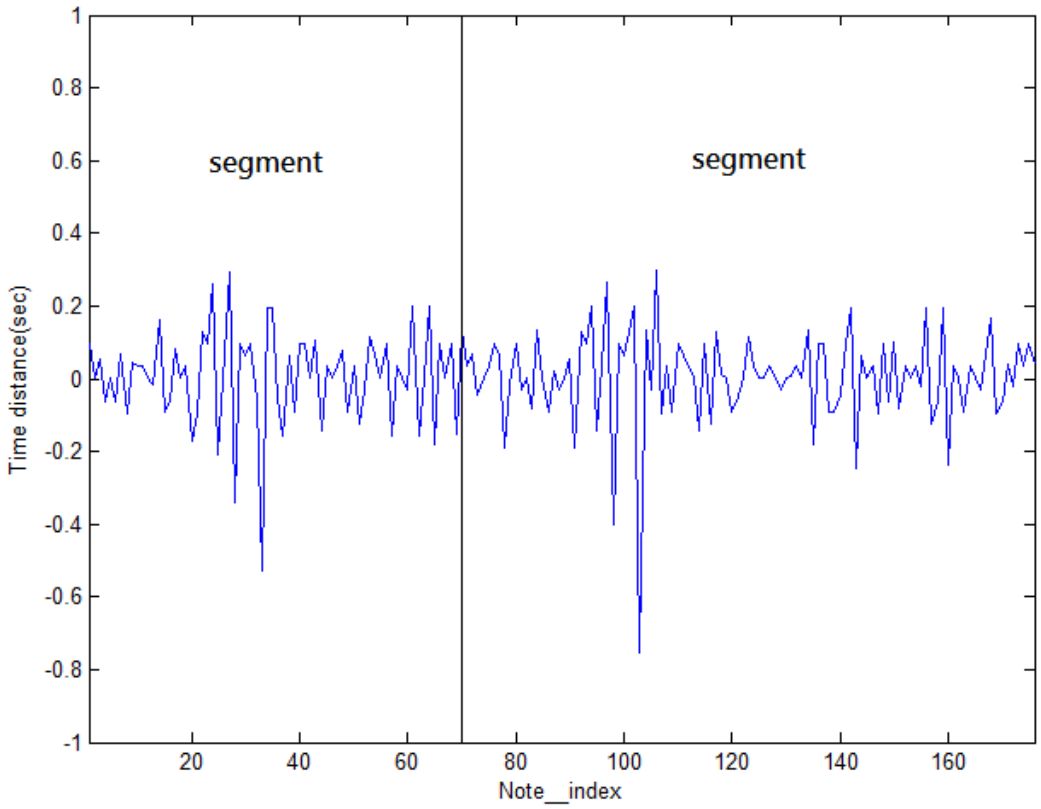


圖 4.12 $\Delta T_r - \Delta T_p$ 的結果

由上圖可以發現，在 Note_index 中發生劇烈變化的位置表示其音符起頭音時間與下一個音符的起頭音時間差距是不自然的，因此可以找出 Note_index 中變化較小的音符，以此音符為基準重新做時間調整。這裡將設定另一個門檻值，並設立以下 3 個條件判斷出可以信任的區間，本論文門檻值設定為 0.15 秒，並找出 Note_index 中連續低於這個門檻值的可信任片段，並以可信任片段的第一個音符為基準，參考原始單音 MIDI 中原本的音符間隔做初步對齊。

1. 由於人工製作的 MIDI 檔可能存在微小的時間誤差，這在 segment 時間長度愈長的情

況下愈常發生，因此只要連續低於門檻值的數量到達 15 個音符即成為一個可信任片段，在此條件下，一個 segment 至少有 1 個可以信任的片段，以這些可信任片段的第一個音符為基準做重新調整，可以改善原先 MIDI 以人工製作時的時間誤差，但當可信任片段個數大於 1 個時，重新調整的動作常常會引發音符重疊的現象，解決此問題的做法是在發生重疊時，我們將會刪除較早出現的音符。

2. 若 segment 中連續低於門檻值的數量只介於 5~14 個音符，將以連續低於門檻值最多的片段當作可信任片段，以此可信任片段的第一個音符為基準做重新調整，也就是對 segment 裡所有的音符做時間位移，找出約略的時間位置。
3. 若 segment 中連續低於門檻值的數量小於 5 個音符，此區間裡的音符將維持未對齊前的時間位置不做任何的調整。

經過以上的步驟可以大致找出 MIDI 所對應的歌曲片段，然而初步對齊的結果準確度可能還存在誤差，本論文將使用上一章 Hsu 的方法所求出的音高曲線當作答案，計算它與初步對齊的 MIDI 的 raw pitch accuracy，差距越小辨識率越高，反之則越低。辨識率的計算方式如下

$$\alpha = \text{Correct Point Number} / \text{All Counted Number} \tag{4.8}$$

其中 Correct Point Number 為 MIDI 的音高值和 Hsu 的音高曲線值皆不為 0 的狀況下，且兩者差距在 1 個半音以內的點數，All Counted Point Number 為 MIDI 的音高值和 Hsu 的音高曲線值皆不為 0 的所有點數。

根據計算出的辨識率 α 使用下式(4.9)對初步對齊的 MIDI 做 $\pm\gamma$ 的時間微調：

$$\gamma = (1-\alpha)\times 10 \tag{4.9}$$

其間的時間間隔為 0.05 秒，每次時間位移的同時會把 MIDI 的調調成對應到 Hsu 的音高曲線平均值，並做±5 個 semitone 的調微調，找出在此時間頻率範圍內辨識率最高的位置，對齊的最後結果如圖 4.13 所示

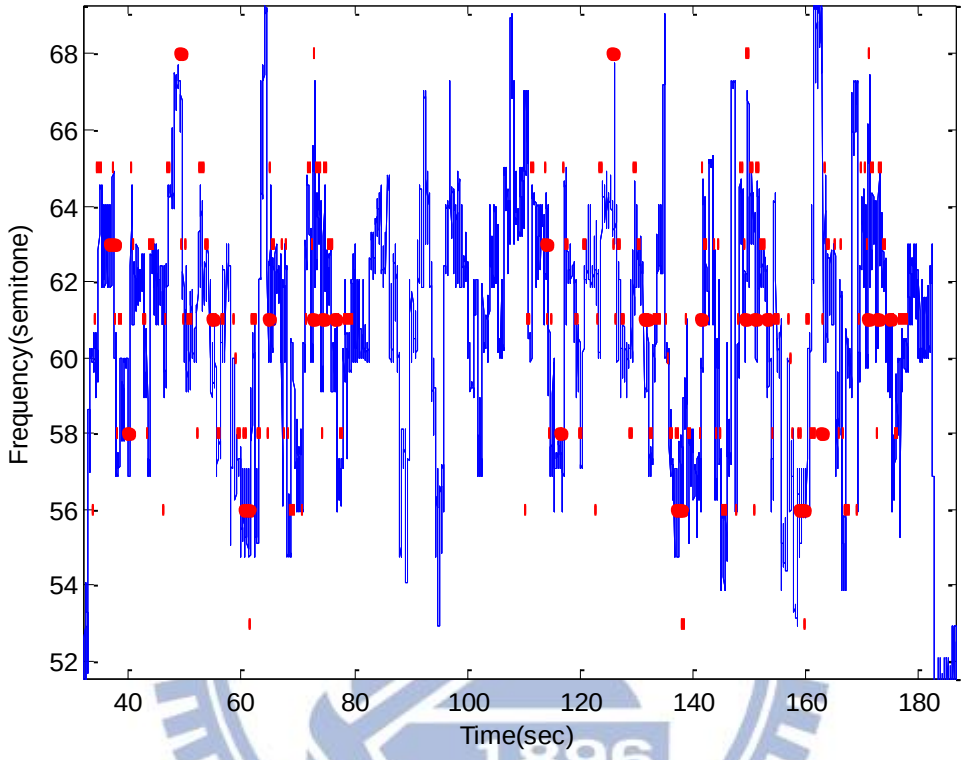


圖 4.13 後處理結果

圖中藍線為 Hsu 的方法所求出的音高曲線而紅線為最後單音 MIDI 對齊的結果

4.3 Pitch Range & Singing Voice Detection

在對齊完單音 MIDI 之後，本論文將延續上一章使用 HPSS 第一階段的 *P* 計算出 NSHS 頻譜，並用對齊好的 MIDI 取代 Hsu 的趨勢估計，這裡設定的範圍為音符所在的音高值上下擴張 2 個 semitone，如此將縮小原本 Hsu 的趨勢估計範圍，並利用單音 MIDI 的特性判斷有人聲段與無人聲段，之後只在有人聲段來做 DP，求出音高曲線，如此即可解決伴奏樂器與人聲銜接時可能產生的音高落差，圖 4.14 為使用對齊後的單音 MIDI

做趨勢估計的結果

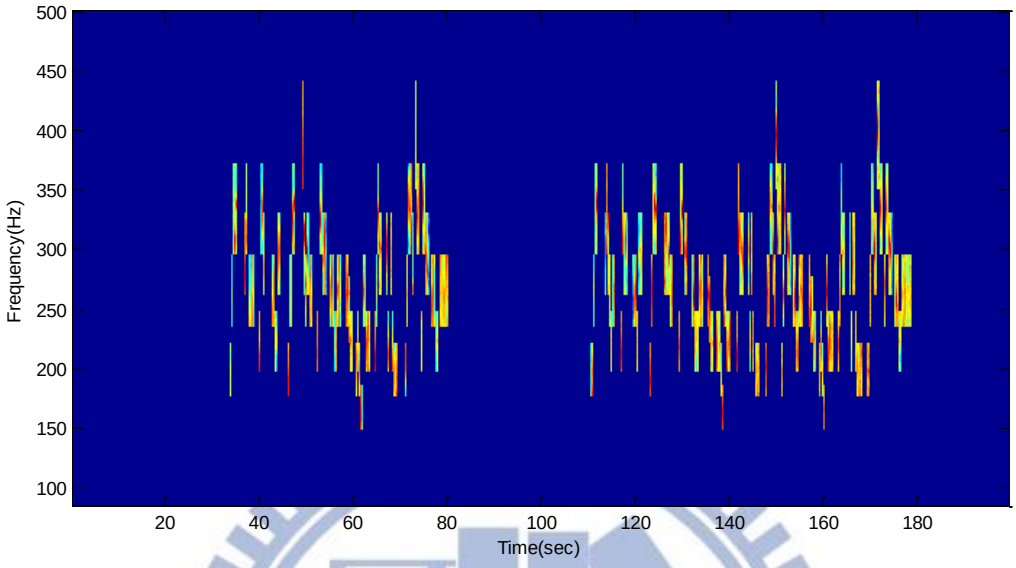


圖 4.14 單音 MIDI 所估計的人聲音高範圍

4.4 DP-based Pitch Tracking

本論文利用圖 4.14 的結果，沿用第三章的 DP 作法來找出最後的音高曲線，其結果為圖 4.15 中黑線部分

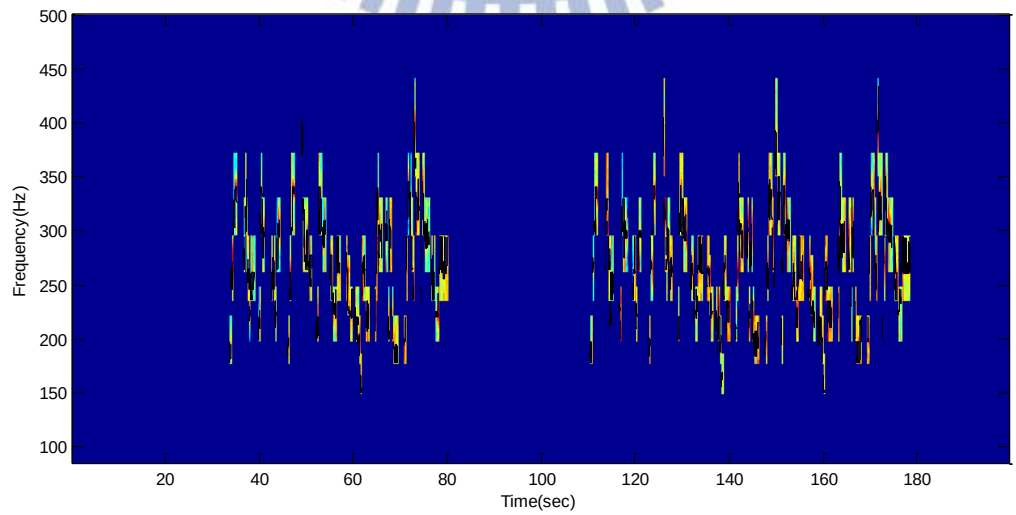


圖 4.15 最後求出的音高曲線

第五章 實驗結果與分析

本章介紹本研究所做的實驗結果並進一步分析結果，5.1 節介紹 MIREX 中各項效能評比；5.2 節描述音高答案的建立；5.3 節列出單音 MIDI 對齊的結果並與 Hsu's 的方法做比較；5.4 節對此系統的辨識結果做分析討論。

5.1 MIREX 中各項效能評比

MIREX 是一個每年都會舉辦，關於音訊訊號處理方面的競賽，競賽裡有各種研究的評比項目，其中有 4 個評比項目是與本研究有相關的，在[14]裡有詳細說明。5.1 式為辨識率的計算公式

$$\text{Recognition Rate} = (\text{Correct Point Number} / \text{All Counted Number}) \times 100\% \tag{5.1}$$

在此，電腦所算出的音高曲線與人工標記答案值為 0 的部分表示為無人聲段，其餘的值皆為音高值，以下說明 4 個評比的計算方法。

Voicing Recall Rate: 在 5.1 式中，Correct Point Number 為人工標記為人聲段且電腦所求出的結果也判斷為人聲段的點數，All Counted Point Number 為人工標記為人聲段的所有點數。

Voicing False Alarm: 在 5.1 式中，Correct Point Number 為人工標記為非人聲段但電腦判斷為人聲段的點數，All Counted Point Number 為人工標記為非人聲段的所有點數。

Raw Pitch Accuracy: 在 5.1 式中，Correct Point Number 為電腦所算出的音高值和人工標記的音高值皆不為 0 的狀況下且兩者差距在 0.5 個半音以內的點數，All Counted Point Number 為電腦所算出的音高值和人工標記答案皆不為 0 的所有點數。由於只比較兩者

皆為人聲段的情況，因此不受人聲段判斷錯誤的影響，而這裡人工標記的答案為使用 MIDI 所做出來的，MIDI 的音高值只會是整數不會有小數，因此錯誤容忍度將放寬到 1 個 semitone。

Overall Accuracy: 在 5.1 式中，Correct Point Number 為人工標記的音高值不為 0 的狀況下，其與電腦所算出的音高值差距在 0.5 個半音以內的點數，All Counted Point Number 為人工標記答案不為 0 的所有點數。此評估計算了系統整體效能，結合了電腦所求出的音高曲線與人聲段判斷的結果，而由於人工標記的答案為使用 MIDI 所做出來的，因此錯誤容忍度也將放寬到 1 個 semitone。

5.2 音高答案的建立

若有純人聲的音檔可以使用一般時域的方法(ex:esps、acf)快速算出音高曲線，並用人工微調做出標準答案。然而在無純人聲音檔的情況下，論文[20]使用了 MIDI 來作為旋律辨識之標準答案，以下介紹建立標準答案的步驟。

1.時間方向的調整

首先使用 Cakewalk 輸入一般音檔(wav)與 MIDI 檔如下圖所示。

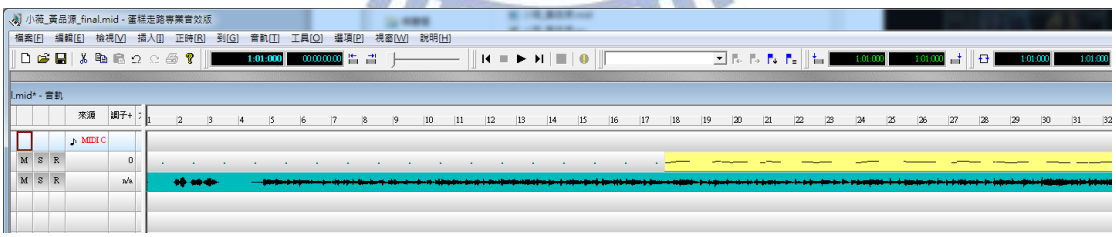


圖 5.1 Cakewalk 範例圖

Cakewalk 是一套專業的 MIDI 製作軟體，它可以消除或建立各種音符，並可以調整其時間位置，在答案製作的過程中會先以人工的方式判斷音符出現的時間是否與音檔一致，並針對不合理的情況來做處理，常見的情況有以下三種

一、有時一個字不一定對應一個音符，有可能是兩個，這種情況下將不會去做修改。

二、若無人聲段的地方出現音符將刪除之。

三、在主歌或副歌為單位的情況下，MIDI 的音符結構不見得會與流行歌曲相同，針對這點將使用 Cakewalk 做部分微調。

調整完後的 MIDI 將用程式來計算出音高曲線，其中的頻率單位為 semitone，時間單位為 0.01 秒，在無音符的地方將標示為 0，之後把此音高曲線使用 wavesurfer 做更細微的調整，其示意圖如下所示

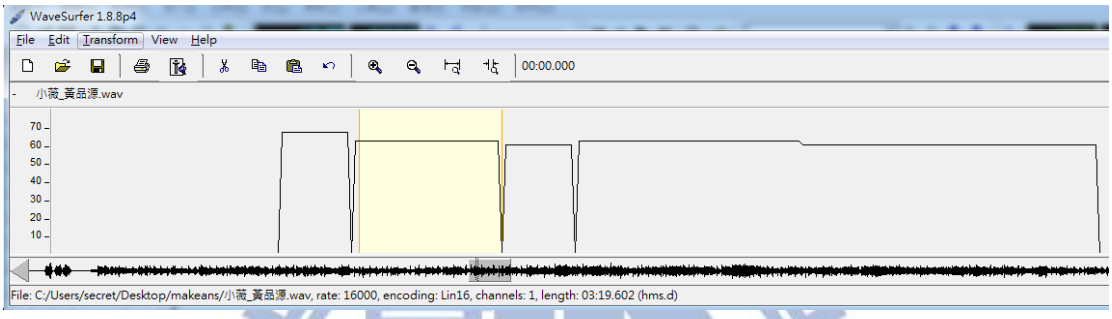


圖 5.2 wavesurfer 範例圖

針對音符多出的部分將使用 wavesurfer 來做修改。

2.頻率方向的調整(key)

以 FIR 的“Lydia”這首歌曲為例，下圖為時間方向調整後的音高曲線答案(紅線)與第三章 Hsu 的方法所求出的音高曲線(藍線)示意圖

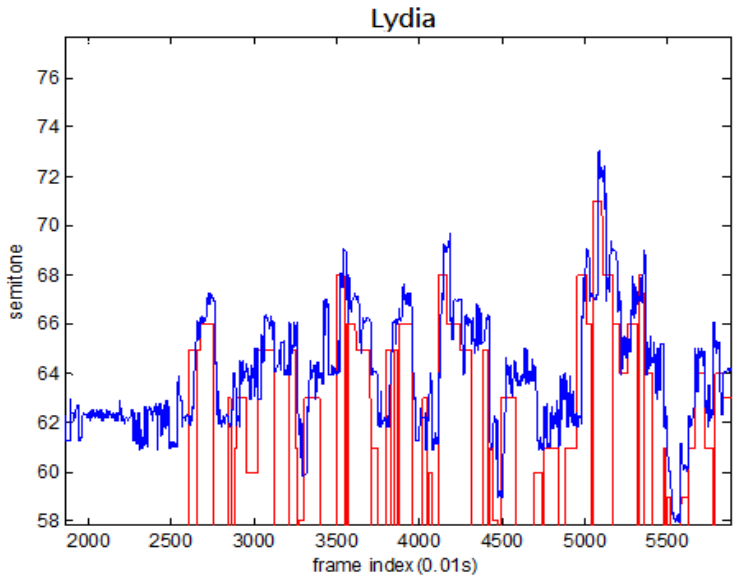


圖 5.3 Hsu 的音高曲線與 MIDI 音高曲線示意圖

由上圖可以發現 MIDI 音高曲線略低於 Hsu 的方法所求出的音高曲線 1 個 semitone，由於 Hsu 的方法已有一定的準確度，因此若差距小於 12 個 semitone 都算合理可以微調的範圍，而範例中針對“Lydia”這首歌曲將把此 MIDI 音高曲線整體上升 1 個 semitone 當最終正確答案。

5.3 Proposed method and Hsu's method

建立完標準答案後，我們可以知道未對齊前的 MIDI 與對齊後的 MIDI 的四種效能評比，圖 5.4 以“外婆的澎湖灣”這首歌為例子，其未對齊前的 MIDI 的四種效能評比 Voicing Recall Rate、Voicing False Alarm、Raw Pitch Accuracy 與 Overall Accuracy 分別為 82.06%、32.25%、12.66%與 10.39%，而對齊後的 MIDI 分別為 91.04%、16.29%、76.08%與 69.26%，可以看到效能都有提升。

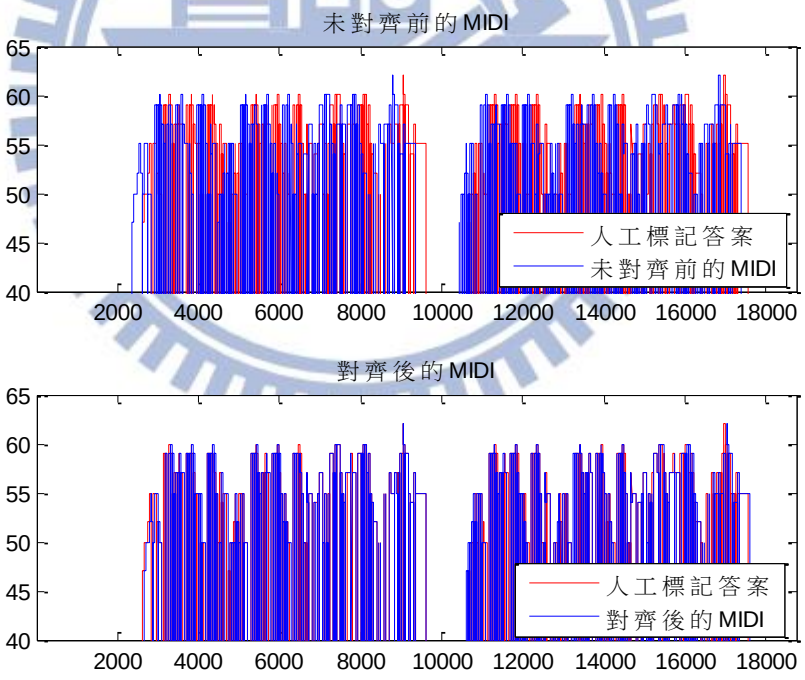


圖 5.4 對齊前後的 MIDI 比較圖

表 5.1 為整體未對齊前的 MIDI 與對齊後的 MIDI 的四種效能評比，可以看到對齊後的 MIDI 整體效能都比未對齊的 MIDI 好，之後將使用對齊後的 MIDI 取代原本 Hsu 的趨勢估計找出頻譜上人聲的頻率範圍，並忽略非人聲段只對有音符的部分做 DP，表

5.2 為 Hsu's 方法與加入 MIDI 改善的音高曲線比較表

表 5.1 未對齊前與對齊後的 MIDI 音高曲線效能評比

	Voicing Recall Rate	Voicing False Alarm	Raw Pitch Accuracy	Overall Accuracy
未對齊前的 MIDI 音高曲線	80.35%	31.97%	20.82%	16.73%
對齊後的 MIDI 音高曲線	92.32%	13.09%	81.76%	75.48%

表 5.2 兩種方法的 Raw Pitch Accuracy 與 Overall Accuracy

	Raw Pitch Accuracy	Overall Accuracy
Hsu's method	64.46%	64.46%
Proposed method	72.23%	66.36%

其中 Hsu 的方法的 Overall Accuracy 與 Raw Pitch Accuracy 相等是因為 Raw Pitch Accuracy 只會考慮兩者皆為人聲段的情況，而 Hsu 的方法並沒有對人聲段與非人聲段做處理，其結果將是一個連續不間斷的音高曲線。圖 5.5 則為兩種方法在各個誤差容忍度下的 Raw Pitch Accuracy。

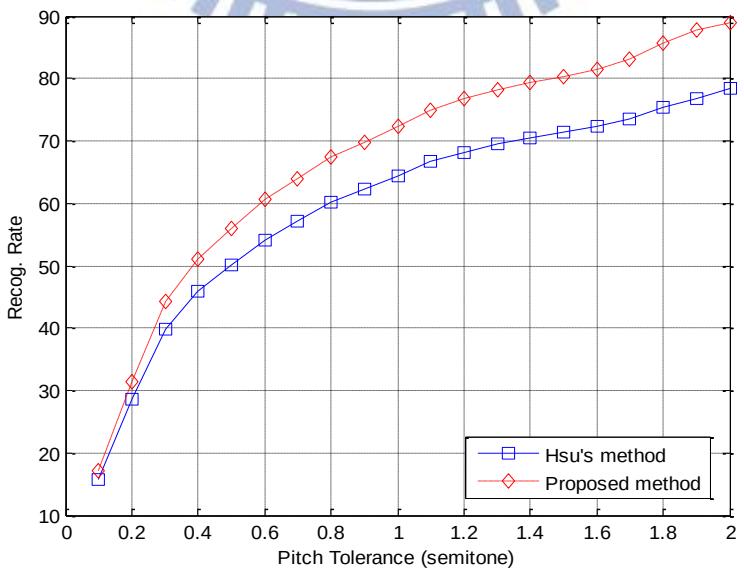


圖 5.5 Hsu's method 與 Proposed method 辨識率曲線圖

圖 5.6 為 Hsu’s method 與 Proposed method 在各個歌曲的 Raw Pitch Accuracy，可以看到除第 6 首歌曲(至少還有你.wav)與第 16 首歌曲(寓言.wav)外，我們的作法都有較高的辨識率。

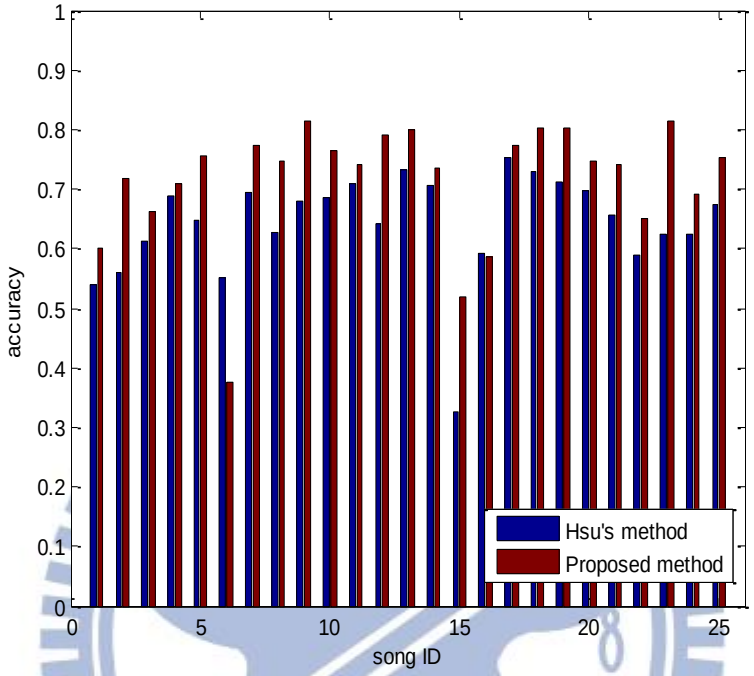


圖 5.6 所收集的流行歌曲其在各方法下的辨識率比較

圖 5.7 為 Proposed method 在各個歌曲的 Raw Pitch Accuracy 與 Overall Accuracy。

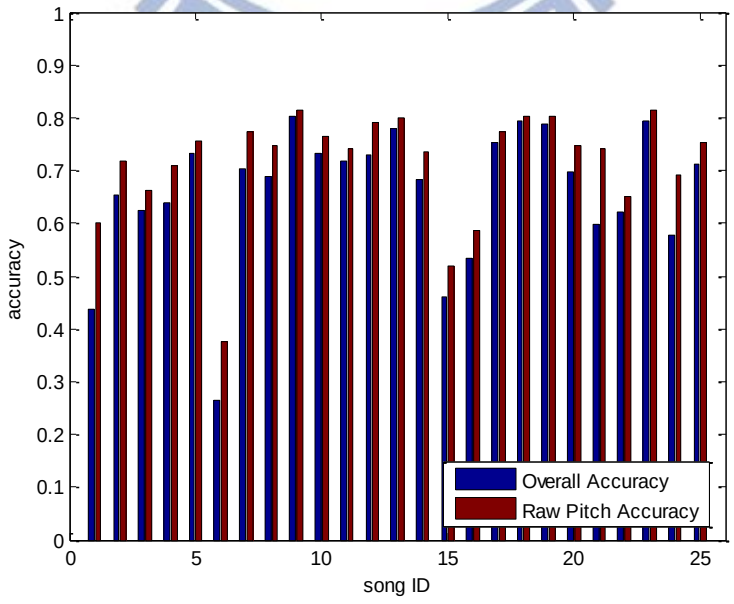


圖 5.7 Proposed method 的音高準確度

表 5.3 為 Proposed method 的 Voicing Recall Rate 與 Voicing False Alarm，可以看到它的辨識率與對齊後的 MIDI 比較都有略低的趨勢，關於這部分將留到 5.4 節做討論。

表 5.3 Proposed method 的 Voicing Recall Rate 與 Voicing False Alarm

	Voicing Recall Rate	Voicing False Alarm Rate
Proposed method	91.89%	12.85%

5.4 辨識結果分析

在 5.3 節中可以看到加入音符資訊後所求出的音高整體來說比 Hsu 的方法還要好，在圖 5.5 中除了第 6 首與第 16 首歌曲之外，各個歌曲都有明顯的進步，其主要歸功於對齊的結果，由於 MIDI 音符紀錄了音高與音長等資訊，且此單音 MIDI 只記錄了人聲主旋律片段，所以只要對齊得夠好，基本上旋律就能完美擷取出來，表 5.4 列出了幾項對齊的主觀結果

表 5.4　Subjective alignment assessment

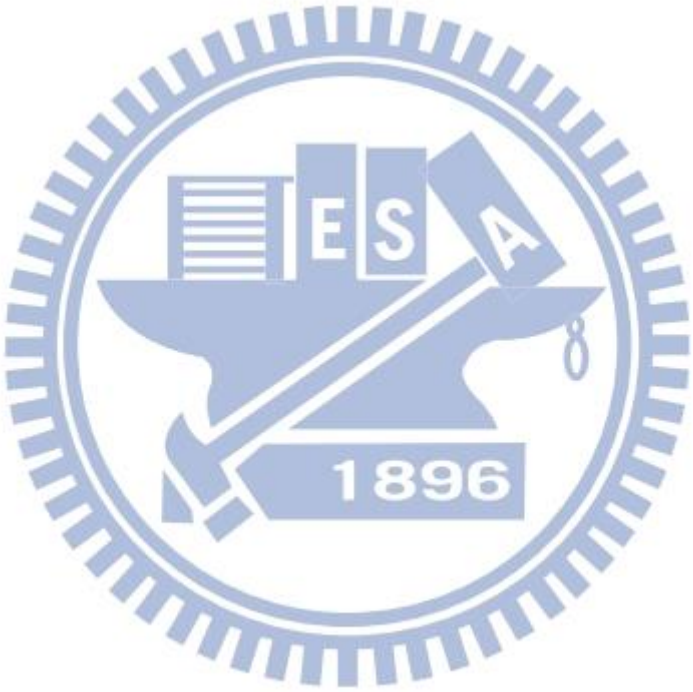
Interpretation	Count / 25
幾乎沒有對齊	1
除了某些主歌或副歌之外，大部分對齊的不錯	5
接近完美的對齊，只有一些小錯誤	5
完美對齊	14

其中幾乎沒有對齊好的歌曲為第六首，因此其旋律擷取辨識率比 Hsu's method 還更差。

使用對齊完後的 MIDI 判斷人聲頻率範圍，進而求取音高曲線，其結果通常會比對齊完後的 MIDI 直接求音高來得合理，主要是因為 MIDI 所紀錄的是音符的資訊，其音高結果是非常平整沒有抖音的現象，即使是受過訓練的專業歌手也不可能每次唱出來的歌曲都符合音符的音高且平整，因此透過對齊完後的 MIDI 來判斷人聲頻率範圍，進而

找出頻譜中能量較大的音高是較合理的。

而在表 5.3 中，Voicing Recall Rate 與 Voicing False Alarm 辨識率會略低於對齊完的 MIDI 主要是因為表 5.3 是使用頻譜所計算出來的音高曲線，而頻譜的時間解析度有其極限，在尋找人聲片段時只能尋找最接近音符時間的音框，其間會有極小的時間誤差，因此當歌曲音符愈多時這極小的誤差就會累積的愈多，而造成辨識率略低的結果。



第六章 結論與未來展望

6.1 結論

本論文主要分為兩個部分，在論文前半部我們使用了準確度相當高的 Hsu's Method 為基礎先求出約略的音高曲線，其方法由於沒有判斷人聲段，音高結果將是一條連續不間斷的曲線，其中也包含了背景伴奏的部分，在求取音高曲線時，背景伴奏與人聲交接處若頻率差距大時，其 DP 的作法不能馬上跳躍到人聲出現的頻率，因而造成 pitch 偵測的錯誤。為了解決這個問題，在第二部份我們引入了單音的 MIDI，它記錄了人聲片段的位置、音高與音長等資訊，然而其音符與對應的流行歌曲位置並不截然相同，因此我們使用了基於 DTW 的方法做對齊，並用一套後處理的方法修正不自然的音符，之後我們用對齊好後的 MIDI 找出頻譜上的人聲片段與人聲頻率範圍，並再一次使用 DP 找出音高曲線。

實驗結果顯示，加入對齊的單音 MIDI 所求出的音高曲線，在 MIREX 的四項評比 Voicing Recall Rate、Voicing False Alarm、Raw Pitch Accuracy 與 Overall Accuracy 中，辨識率分別為 91.89%、12.85%、72.23%、66.36%，與 Hsu 的方法比較，Raw Pitch Accuracy 評比改善了 7.77%絕對錯誤率，且多了"有人聲段"的判斷結果。

6.2 未來展望

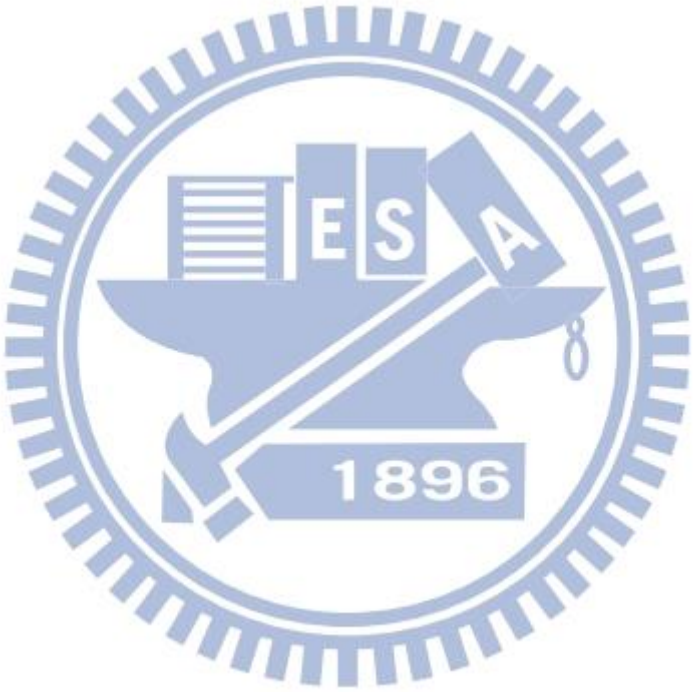
從本研究可延伸出以下幾個議題值得在未來探討：

一、改善 MIDI 的對齊時間：由於 DTW 的複雜度相當高，若音框太小歌曲長度又長，其計算時間將相當可觀；

二、背景伴奏的消除：以目前的結果來看，HPSS 的處理效果離純人聲還有一段距離，背景伴奏有如雜訊，常造成後續音符對齊與旋律擷取的錯誤；

三、MIDI 音符的來源：本研究所使用的 MIDI 是從網路上所取得的，它只記錄了人聲

主旋律的部分，而大部分的 MIDI 為多音的 MIDI，除了人聲主旋律部分外還有背景伴奏，未來可以針對此部分做探討。



參考文獻

[1] A. Ghias, J. Logan, D. Chamberlain and B. C. Smith, “Query by humming-musical information retrieval in an audio database”, ACM Multimedia ’95, San Francisco, 1995.

[2] M. Goto, “A Real-Time Music Scene Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals”, Speech Communication, vol.43, no. 4, pp.311–329, 2004.

[3] C.-L. Hsu and R. Jang, “SINGING PITCH EXTRACTION AT MIREX 2010”, The Music Information Retrieval Evaluation Exchange, 2010.

[4] C. Raphael, “Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models”, IEEE Trans. on PAMI, vol.21, pp.360-370, 1999.

[5] R. Turetsky and D Ellis, “Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI syntheses”, Proc. Int. Symp. Music Info. Retrieval, Baltimore, 2003.

[6] MIDI 製作過程, http://v.youku.com/v_show/id_XMTc5MDA3OTU2.html

[7] YouTube to mp3, http://www.youtube-mp3.org/

[8] H. Tachibana, T. Ono, N. Ono and S. Sagayama, “Melody line estimation in homophonic music audio signals based on temporal-variability of melody source”, IEEE ICASSP, pp.425-428, 2010

[9] D. J. Hermes (1988), “Measurement of pitch by subharmonic summation”, J. Acoust. Soc. Am. 83, 257-264.

[10] K. Dressler, “Sinusoidal extraction using an efficient implementation of a multi-resolution FFT”, DAFx, pp.247–252, 2006.

[11] M. DONG, P. CHAN, L. CEN and H. LI, “Aligning Singing Voice with MIDI Melody Using Synthesized Audio Signal”, ISCSLP, pp.95–98, 2010.

[12] K. Schutte , http://www.kenschutte.com/midi

[13] H.-M. Yu, W.-H. Tsai and H.-M. Wang, “A query-by-singing system for retrieving karaoke music”, IEEE Trans. on Multimedia, 10 (8), pp.1626–1637, 2008.

[14] G. E. Poliner, D. P. W. Ellis, F. Ehmann, E. Gómez, S. Steich and B. Ong, “Melody transcription from music audio: Approaches and evaluation”, IEEE Trans. on Audio, Speech, and Language Processing, vol. 15, no.4, pp.1247–1256, May 2007.

[15] J.-S. R. Jang, “Audio Signal Processing and Recognition”, (in Chinese) available at the links for on-line courses at the author's homepage at http://www.cs.nthu.edu.tw/~jang.

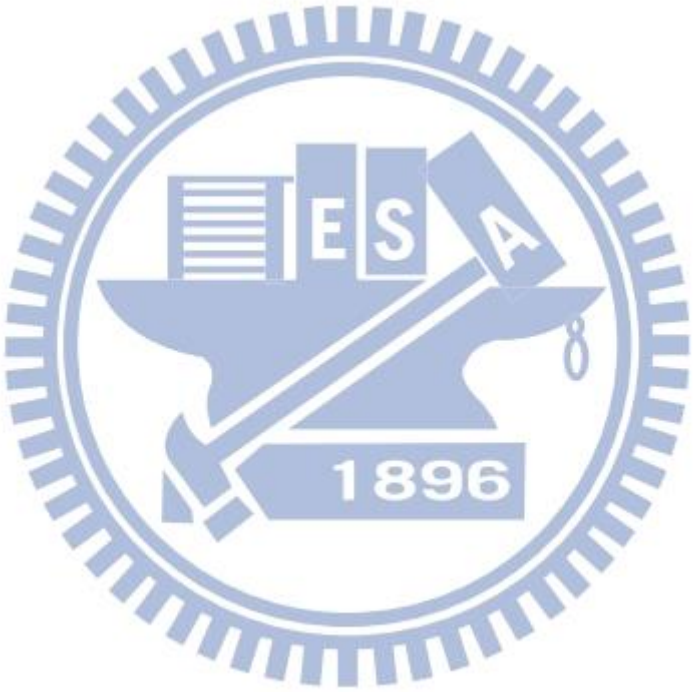
[16] D. P. W. Ellis (2008). "Aligning MIDI scores to music audio", web resource: http://www.ee.columbia.edu/~dpwe/resources/matlab/alignmidiwav/

[17] 白宗儒, “一個適用於複音音樂之音高追蹤的混成法”, 國立清華大學碩士論文, 2011

[18] 林冠延, “使用色度特徵的複音音樂訊號分析與檢索”, 國立成功大學碩士論文, 2009

[19] 李宏儒、許肇凌、王儀蓁、張智星, “多模式音樂檢索系統”, 第三屆數位典藏技術研

討會，中央研究院, Taiwan, August 2004.
[20] 鄭琇云，“各種音高追蹤方法對哼唱選歌之影響的評估”，國立清華大學碩士論文，2008



附錄:歌曲資料庫

Son ID	歌曲名稱	歌手名稱
1	Lydia	FIR
2	honey	王心凌
3	小薇	黃品源
4	外婆的澎湖灣	潘安邦
5	如果這都不算愛	張學友
6	至少還有你	林憶蓮
7	後來	劉若英
8	星光遊樂園	twins
9	星語心願	張柏芝
10	為什麼你背著我愛別人	許志安
11	值得	鄭秀文
12	莫斯科沒有眼淚	twins
13	都是夜歸人	許美靜
14	最初的夢想	范瑋琪
15	最熟悉的陌生人	蕭亞軒
16	寓言	張韶涵
17	開不了口	周杰倫
18	黃昏	周傳雄
19	當你孤單你會想起誰	張棟梁
20	當愛在靠近	劉若英
21	零	柯有倫
22	遙遠的等待	江惠
23	親愛的你怎麼不在我身邊	江美琪
24	簡單愛	周杰倫
25	藏鏡人	秦揚